A Work Project, presented as part of the requirements for the Award of a master's degree in business Analytics from the Nova School of Business and Economics.

The Relationship between Programming Literacy and Economic Growth in Europe

Analysis of Programming Literacy and Economic Growth in Poland (2008-2020)

Aleksandra Taraszka - 53456

Work project carried out under the supervision of:

Michael Kummer

20-12-2023

Abstract

The thesis investigates the link between programming literacy and economic growth across European regions, aiming to understand their interplay more comprehensively. Using statistical models, the research predicts the economic effects of programming literacy, revealing positive correlations between literacy and economic prosperity. The study also presents a nuanced case study on Poland, emphasizing the need for a tailored approach due to the country's unique development stage in comparison to other European countries.

Keywords

[Programming Literacy, Economic Growth, Stack Overflow, NUTS-3 Regions, Poland]

This work used infrastructure and resources funded by Fundacao para a Ciencia e a Tecnologia (UID/ECO/00124/2013, UID/ECO/00124/2019 and Social Sciences DataLab, Project 22209), POR Lisboa (LISBOA-01-0145-FEDER-007722 and Social Sciences DataLab, Project 22209) and POR Norte (Social Sciences DataLab, Project 22209).

Table of Content

List of Figures

List of Tables

1 Introduction

In today's digital age, the importance of programming literacy in driving economic growth cannot be understated. This Thesis examines the intersection between programming skills and economic development within Europe, with a specific focus on analyzing data at a granular level. By shedding light on how programming proficiency impacts regional economies, this research aims to bridge a critical gap in our understanding of how digital competencies contribute to overall prosperity.

As European nations transition into "new economies" that heavily rely on digital capabilities, it is essential to explore the transformative role of programming literacy. These once niche skills have now become indispensable for sustainable economic growth—comparable to traditional literacy during previous centuries. Through active participation in platforms like Stack Overflow, individuals demonstrate their fluency in coding languages as well as their ability to problem-solve collaboratively—an aspect crucial for thriving in today's interconnected world.

This study seeks answers through several interrelated questions:

- What factors contribute to the emergence of programming hubs, influencing the distribution and growth of programming literacy across diverse regions?
- To what degree does the concentration of programmers in specific regions correlate with economic indicators such as GDP, GVA, and employment rates?
- Can economic factors, such as GDP and employment, be considered significant predictors of programming literacy in specific regions?

The aim of our research is twofold: to examine the relationship between digital skills and regional economic growth, and to provide insights for policymakers to foster environments that nurture these skills.

Employing a comprehensive methodological approach, the thesis analyzes a confluence of programming activity and economic data. It utilizes advanced statistical models including OLS,

Ridge, Lasso, Random Forest, and ARIMA for outlier analysis, aiming to predict and understand the economic impact of programming literacy. This approach not only assesses the direct correlation between programming skills and economic factors but also explores the predictive capability of these variables in forecasting regional economic trends.

This research contributes significantly to our understanding of how programming literacy impact regional economies. By identifying key economic factors that influence programming literacy levels and assessing their potential for forecasting purposes, this study offers valuable insights into strategic decision-making during Europe's ongoing digital transformation. The findings are particularly relevant for policymakers seeking to shape educational curricula aligned with industry needs as well as business leaders aiming to leverage coding expertise as a driver of sustainable economic development.

The paper's structure includes introductory chapters, methodology explanation, data exploration of programming literacy on country level as well as on NUTS-3 level. A regression model analysis for predicting programming literacy based on economic metrics follows. The study delves into outlier analysis and includes five additional papers. These include a panel data analysis to compare the efficacy of linear and machine learning models in economic forecasting, a case study on Poland examining the nexus of programming skills and economic growth, and an analysis of the creation and growth of programming hubs within Poland. The study also contemplates how the inclusion of varied economic and technological factors could refine predictive models. The concluding chapter synthesizes these discussions, probing the potential relationship between programming literacy and economic resilience, thereby providing strategic insights into the sustenance and fortification of regional economies in the face of adversities.

2 Theoretical Foundations

2.1 Definitions

In this subchapter, we define relevant terms to clarify their scope within the context of this paper, ensuring that their meaning is precisely communicated and understood.

Programming Literacy: Balanskat and Engelhardt (2014) define computer programming as „the process of developing and implementing various sets of instructions to enable a computer to perform a certain task, solve problems, and provide human interactivity“ (21). However, when referring to programming literacy, the term expands beyond a mere technical skill limited to few specialists. We define it as a comprehensive understanding of computational processes, a widely held ability integral to communication and societal interaction in the digital age, reflecting its extensive applications and significance (Vee 2013, 46-47).

Economic Growth: Simply put, economic growth refers to “an increase in the quantity and quality of the economic goods and services that a society produces” (Roser 2023). Through this process economic well-being and material quality of life are improved (Investopedia 2023). Various metrics to measure economic growth exist. For the scope of the thesis, our dataset focuses on population size, Gross Domestic Product, Gross Value Added and employment rates.

- *Gross Domestic Product (GDP):* GDP is the total monetary value of all goods and services produced within a country's borders in a specific time period. It encompasses not only the income generated from production but also the aggregate expenditure on final goods and services, accounting for the deduction of imports. However, GDP has limitations for temporal comparisons, as its variations over time may be attributed not only to actual

economic growth but also to fluctuations in prices and purchasing power parities (PPPs). (OECD n.d.-a)

- *Gross Value Added (GVA):* GVA is an economic metric that quantifies the value of goods and services produced in a specific area, industry, or economic sector. It is calculated by subtracting the cost of intermediate consumption from the total output value, representing the net contribution of labor and capital in the production process. This measure also indicates the income generated from these contributions. GVA by activity highlights the unique value added by different sectors like agriculture, industry, and services, often expressed as a percentage of the economy's total value added. (OECD n.d.-b)
- *Employment rates:* Employment rates are indicators that assess the utilization of available labor resources, specifically individuals who are ready and able to work. These rates are determined by calculating the proportion of employed individuals relative to the population within the working age bracket, typically defined as ages 15 to 64. The ratio serves as a gauge for understanding how effectively a society is employing its potential workforce. (OECD n.d.-c)

Nomenclature of Territorial Units for Statistics (NUTS): To measure geographical developments of programming literacy across Europe, our dataset works with a Eurostat classification known as NUTS (Eurostat 2022). This classification divides regions within the European Union using a hierarchical, three-level system based on population sizes. The purpose of this system is to provide a single, uniform breakdown for the collection, development, and harmonization of EU regional statistics. Table 1 captures the 3-level categorization. Each level is designed to provide a standardized comparison between the areas. By utilizing the NUTS classification, our research can achieve a more granular understanding of how programming literacy varies not only between countries but also within different regions of a single country.

Table 1 Categorization of the Nomenclature of Territorial Units for Statistics Regions

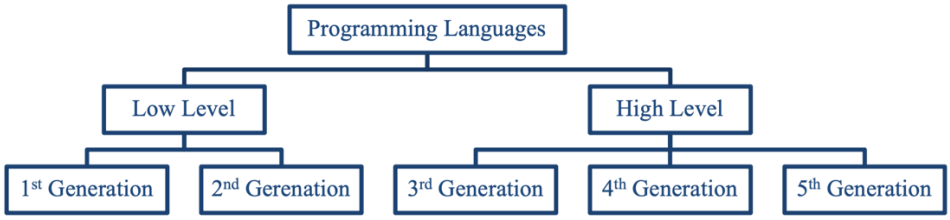
NUTS-1	These are major socio-economic regions. In larger countries, these might be large regions or groups of federal states, while in smaller countries, it can be the country itself. The population of each NUTS-1 region is typically between 3 million and 7 million people.
NUTS-2	This level corresponds to basic regions for the application of regional policies. They can be provinces or smaller states within a country. The aim is to provide a regional breakdown that is consistent for economic analyses. NUTS-2 regions usually have populations between 800,000 and 3 million people.
NUTS-3	These are smaller regions, often for more local administrative purposes, such as counties or larger urban districts. The NUTS-3 level is used for more detailed regional planning and comparisons. These areas usually have populations ranging from 150,000 to 800,000 people.

Stack Overflow: Stack Overflow is a question-and-answer platform for knowledge sharing across programming communities. The platform serves as a repository of information about computer programming. Stack Overflow facilitates collaboration, problem-solving, and knowledge-sharing among individuals, groups, and organizations (Sartore 2022). Established in 2008, Stack Overflow has emerged as a digital forum for programmers worldwide. The regional activity on Stack Overflow is going to serve as our primary metric of programming literacy.

2.2 History of Programming Languages

GitHub's latest October report stated that in 2022 almost 500 primary languages were used on their platform to build software (GitHub 2023). However, the history of programming languages spans over a century of innovation, reflecting the evolving needs of computation, industry practices, and academic research. An effective approach to illustrate these major advancements is by categorizing them into five distinct generations, each characterized by increasing levels of abstraction and sophistication (see Figure 1).

Figure 1 Overview Programming Languages Generations



Note. Adapted from *Generation of Programing Languages* by GeeksforGeeks, 2023. Retrieved from https://www.geeksforgeeks.org/generation-programming-languages/

1st Generation Languages (1940s-1950s) – Machine Languages: The advent of programming languages can be traced back to the 1940s and 1950s with the development of machine languages, also called low-level languages, the most fundamental form of programming languages (CSDH 2022). These first-generation languages are characterized by their use of binary code, directly executable by the computer's central processing unit (CPU), making them fast and efficient with no translator needed (GeeksforGeeks 2023).

2nd Generation Languages (1950s-1960s) – Assembly Language: Emerging in the 1950s and 1960s, the second generation of programming languages introduced assembly languages. These languages represented a significant step forward, utilizing mnemonic codes that were more accessible than the binary code of the first generation. Assembly languages allow programmers to write instructions in symbolic code, which an assembler would then convert to machine code. (Medewar n.d.; GeeksforGeeks 2023)

3rd Generation Languages (1960s-1970s) – High-Level Languages: The third generation of programming languages, emerging around the 1960s, marked a significant evolution in the field, as they are machine-independent, meaning one can run the same code on different machines (Jain 2018). However, this means that for translating them into machine language a compiler is needed, and different machines use different compilers. Therefore, developers must use compilers accordingly (Medewar n.d.). These high-level languages, such as FORTRAN,

PASCAL, ALGOL, and COBOL, shifted closer to human language in their syntax, including English-like keywords and hence greatly enhancing the efficiency and accessibility of programming (Chandra n.d.; Jain 2018).

4th Generation Languages (1970s-1980s) – Very High-Level Languages: The fourth generation of programming languages, arising in the 1970s, focused on further increasing programmer productivity and efficiency. These very high-level languages, exemplified by SQL and MATLAB, are often domain-specific, targeting areas such as database management and report generation as well as scientific computation (Chandra n.d.; GeeksforGeeks 2023).

5th Generation Languages (1980s-Present) – Artificial Intelligence Languages: The fifth and most recent generation of programming languages make use of visual tools to help develop a program. A popular example for this is Visual Basic. This generation centers on artificial intelligence (AI) and constraint programming. This approach is particularly prevalent in AI and machine learning research. While 5th generation languages are crucial in these advanced fields, their use is not as widespread in general-purpose programming, reflecting their specialized nature. (Jain 2018)

6th Generation? – In the forthcoming years, we expect programming languages to evolve in response to technological advancements, industry challenges, growth of automation, AI and machine learning as well as the increased need for collaboration due to increasing (semi-) remote work and lastly economic considerations (Schaefer 2023). The next decade will likely see a trend towards specialization, with programming languages tailored to unique domains such as quantum computing and augmented reality.

The retrospective of programming languages, from their genesis in machine code to the sophisticated AI and domain-specific languages of today, sheds light on the exponential

trajectory of programming literacy. It is a journey marked by the relentless pursuit of efficiency, abstraction, and the bridging of the human-computer divide. As each generation of languages abstracted complexity and made programming more accessible, the skills to harness them became increasingly prevalent. Today, programming literacy is not only widespread but also integral to innovation and economic development, reflecting the demand for digital competency in an increasingly tech-driven world. The future promises even greater accessibility, with languages evolving to meet the needs of new technological frontiers.

2.3 Literature Overview

In the contemporary global economy, the role of programming and digital skills, collectively referred to as Information and Communications Technologies (ICTs), has emerged as a critical factor influencing economic growth. Numerous studies worldwide have explored the correlation between digital skills and the economic development of specific countries. Global research consistently confirms a correlation between the development of digital skills and the economic growth of nations. Scholars have explored this linkage using diverse methodologies and frameworks (Sein et al. 2018).

For instance, Laitsou, Kargas, and Varoutas (2020) conducted a comprehensive study spanning a 20-year period, with a special focus on the economic crisis period (2008–2016) in the European Union. Using growth accounting methodology and regression analysis the study revealed that investments in ICT, especially during economic crises, play a pivotal role in fostering economic growth in the Eurozone, and specifically growth of GDP in Greece in the investigated timeframe. This suggests the relation between economic growth and programming literacy this paper investigates.

Urbančíková, Manakova, and Bielcheva (2017) delved into the European context, specifically focusing on the impact of digital skills on the local economy in Slovakia. They assessed the association between education, income and household type and digital skills. The paper proved that those factors are significantly affecting digital literacy of the society, rather than the size of the economy or the economic growth of the technological industry in the country.

Building on this, Bălăcescu (2019) investigated the relationship between economic growth and digital skills across EU member states. Using GDP per capita and digital skills, defined as BOSE index (basic overall digital skill), the research identified similarities and dissimilarities among EU countries. The findings provide valuable insights for decision-makers, proving that development of digital skills can hinge the growth of European economies.

As evidenced by the research, the symbiotic relationship between programming literacy and economic growth is clear, with digital skills serving as both a catalyst for and an indicator of a nation's economic vitality. Our study builds upon this foundation to empirically explore the specific correlation between programming literacy and economic development in NUTS-3 regions of six European countries.

2.4 The Rise of European Tech Hubs

In 2019, Germany led the European landscape with over 901,400 professional developers, followed by the UK with around 849,600. Other significant contributors included France, Italy, the Netherlands, and Poland (Statista 2019). Highlighting Europe's emergence as a technological leader, recent data reveals a significant and sustained rise in its developer workforce. With around 6.1 million professional developers in 2019, Europe has overtaken the United States in this metric. This marked growth in developer populations across European regions not only cements Europe's position at the forefront of global tech innovation but also

mirrors the burgeoning concentration of tech professionals within rapidly growing tech hubs (Atomico, Slush, and Orrick 2019).

Central to this development has been the role of government policies and incentives. European governments have increasingly recognized the importance of the digital economy, enacting policies that encourage innovation and entrepreneurship. The European Commission's "Digital Single Market" strategy, for instance, aims to open up digital opportunities for people and businesses and enhance Europe's position as a world leader in the digital economy, introducing migration friendly policies to attract global talent (European Commission 2013). Such policies create an environment conducive to the growth of tech hubs by providing the necessary support structures for startups and established tech companies alike. Government regulations, digital literacy campaigns, and infrastructure investments have been instrumental in establishing conditions conducive to technological innovation. Further insights are provided by the EU Innovation Scoreboard's 2021 "Annual Report on European Tech Hubs," particularly regarding the post-COVID-19 landscape. The report emphasizes innovation's role in tech hub evolution and notes how the European Commission's updated industrial strategy, focusing on SMEs and startups, is reshaping the technological landscape.

Hence, availability of venture capital and other forms of investment is another key driver for the growth of tech hubs. This influx of capital acts as a magnet for these startups, which are key to drawing in a pool of skilled programmers. Notably, hubs with dense concentrations of investors, especially those with a keen interest in cutting-edge technologies, often coincide with areas rich in research and development institutions. These include universities, such as the ETH in Zurich, tech incubators, and specialized medical facilities. (Davis et al. 2023) These factors combined are prevalently given in urban regions of economically developed countries, such as Berlin, London, or Stockholm. Finally, soft factors, as quality of life and cultural diversity

additionally impact a regions attractiveness to developers, enriching its location with programming expertise (Varley 2023).

3 Methodology

In the forthcoming analysis, we will conduct an empirical examination of programming activity across six European countries, applying quantitative analyses. By integrating economic indicators with programming literacy data, we aim to explore the extent to which a country's economic development serves as a predictive measure for its programming rates, thus validating theoretical assertions and shedding light on the interplay between economic vitality and a population's programming skills. Rigorous data collection from sources such as Stack Overflow and Eurostats, analyzed using Python, will help address potential biases and ensure a robust exploration of the relationship between programming literacy and economic factors in European regions.

3.1 Main Dataset

The main dataset contains information on programming activity and economic factors over the years by NUTS-3 region. Regarding programming activity, Stack Overflow metrics *question, answer, comment, upvote,* and *downvote counts* were used. Economic indicators include *employment, gross value added (GVA), gross domestic product (GDP)* and *population*, covering the countries Spain, France, Italy, Turkey, Sweden, and Switzerland.

3.2 Data Sources

The data on Stack Overflow activity by NUTS-3 regions is provided by Stack Overflow. The economic data is curated from reliable online sources, such as Eurostats and national statistics databases. Merging additional data requires an identifier for accurate integration. NUTS-3

names from the main dataset are used for this purpose. Discrepancies with Eurostats data, which uses NUTS-3 identification codes, require matching the regions with corresponding codes. An official Excel sheet from Eurostats that maps the regions to their codes facilitates this process. Differences in region names between the main and Eurostats datasets are solved by creating a mapping file, ensuring no data loss during linkage. Economic factors are chosen according to the ones that are defined as regional economic accounts by the European Commission (ESA 2010), namely employment, gross domestic product (GDP) at current market prices, gross value added (GVA) at basic prices and the average annual population to calculate regional GDP data. After merging this data with the main dataset, a subsequent search for missing values reveals significant gaps in economic data for Switzerland and Turkey. Missing data for Switzerland is collected from the Federal Statistics Office (FSA) and for Turkey from the Turkish Statistical Institute. The further handling of missing data is discussed in the following chapter.

3.3 Data Preprocessing

Data collected from various sources, especially in a multifaceted field that combines economic factors and programming activities across NUTS-3 regions, is often raw, unstructured, and may contain inconsistencies or errors. Data cleaning is an essential step to transforming this raw data into a clean, structured format suitable for analysis.

Given the temporal and regional diversity of the dataset, ensuring comparability across different time periods and regions is crucial. Preprocessing standardizes the data, allowing for meaningful comparisons and trend analysis. Moreover, in real-world datasets, the presence of missing values is a common challenge. It is necessary to implement strategies to handle missing data, either through imputation or removal, depending on their impact on the overall analysis.

In particular, tackling the challenge of rows with missing years, the most relevant time period for analysis is investigated, considering the focus on programming activity. This approach allows for the possibility of excluding certain years where data is unavailable.

Figure 2 Stack Overflow Activity across the Years based on ES, IT, FR, SE & CH data

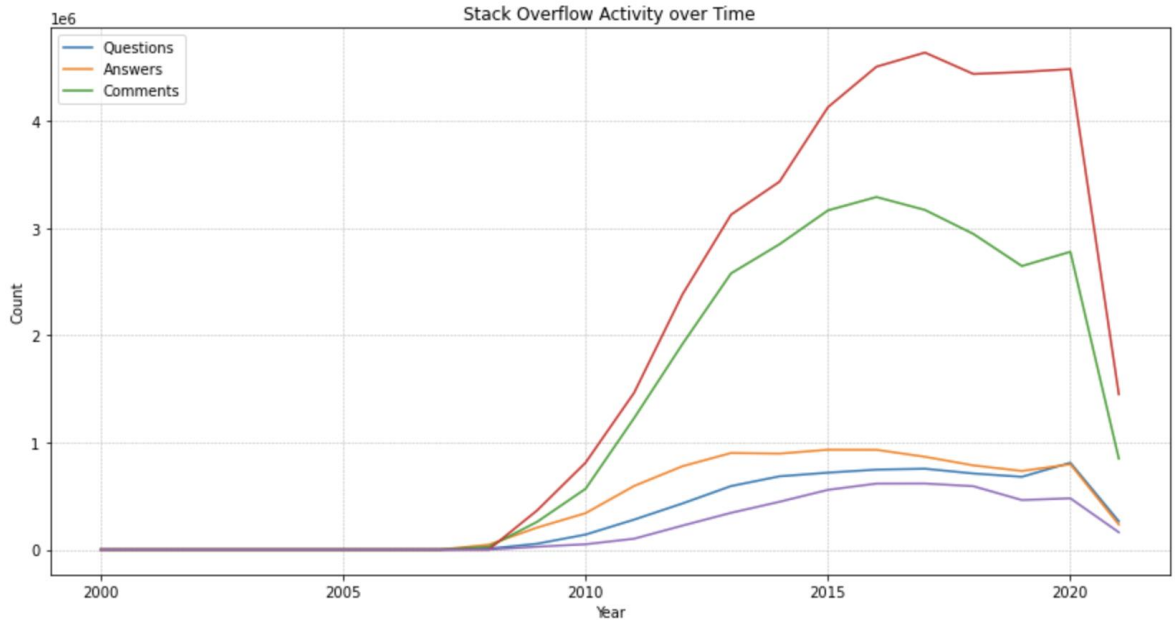


Figure 2 shows different Stack Overflow activities plotted over time. The Figure reveals no data before 2008 and a notable decrease in 2021, suggesting the data collection occurred within that year, omitting complete data for 2021. Consequently, the analysis period is set from 2008 to 2020. This adjustment significantly reduces the number of missing values for the economic data. However, gaps remain, notably in employment data for Switzerland and Turkey, and in GDP, GVA, and population data for Switzerland.

Handling missing data for Switzerland involves solving several different problems. First, to align with Eurostats' average annual population data for other countries, it is necessary to calculate the average yearly population for each Swiss canton. These calculations match the values for Switzerland in the Eurostats data. Second, the Swiss data for GDP and GVA, presented in millions of Swiss francs, differs from the other countries' data in millions of euros. To resolve this, the average CHF/EUR exchange rate per year, provided by the European

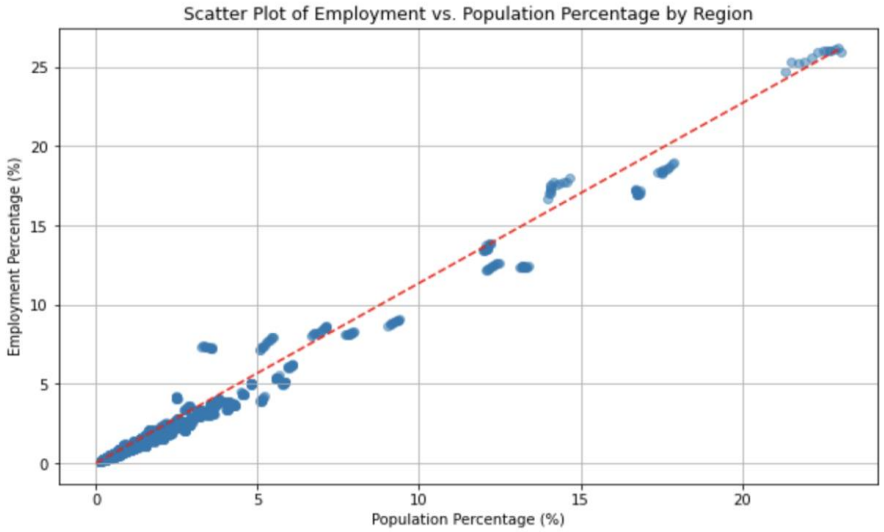
Central Bank Data Portal, is used to convert Swiss values into euros. Last, employment data is only available from 2010 onwards, resulting in a lack of information for the initial two years (2008 and 2009) of the defined analysis period. These values are backwards filled with the respective observations of the year 2010.

Regarding employment data for Turkey, it is not possible to find any reliable data that is grouped by NUTS-3 regions. The granularity of the data provided by the Turkish Statistical Institute is monthly and on a national level. The monthly data can easily be transformed by calculating the yearly average, but the fact that the data is only on a national level imposes a more difficult problem.

It is hypothesized that the distribution of employed persons per year per NUTS-3 region relative to the national level is similar to the distribution of the population. Thus, missing data can be calculated by using this distribution. To test this hypothesis, it is necessary to show that this holds true for the other five countries in the dataset. A statistical analysis is performed, and the distributions are plotted against each other.

In Figure 3 the percentages of employment and population by regions relative to the countries are plotted against each other. By observing the relationship between the distribution of employment and population across the different regions one can infer that there is a high positive correlation which suggests that regions with a high population density tend to have high employment rates. The red dashed line represents the line of equality, which means that it's the point where both distributions are equal. Most of the points lay very close to the line. Additionally, a Pearson Correlation Coefficient of 0,989 is calculated. This further suggests a very highly positive correlation between both indicators. In other ways, as the population increases in a NUTS-3 region, the employment rate tends to increase linearly and vice versa.

Figure 3 The Relation between Employment and Population in Turkey



However, there are concerns as to whether it makes sense to fill the values in this way. Firstly, the underlying assumption of a uniform relationship between population and employment across all regions could be problematic because variations in employment patterns due to economic, social, or geographical factors might not be captured. This means future models might oversimplify the employment landscape. Secondly, outliers in the data are important because they can represent regions with unique employment characteristics. For instance, a region with a small population but high employment due to a major industrial center would be an outlier. By aligning employment strictly with population, future models might miss these nuances. The outliers, rather than being anomalies to be smoothed over, could be essential for understanding regional economic dynamics. Lastly, the relationship between population and employment is not static. It can change over time due to various factors like economic policies, technological advancements, and social changes. The assumption that past relationships will hold in the future can be problematic, especially in regions undergoing rapid change. Generally, if population and employment are highly correlated, using both as separate predictors in a regression model could lead to multicollinearity, where it becomes difficult to separate the

effect of one predictor from the other. This can inflate the variance of the coefficient estimates and make the model less reliable.

To address these concerns, it is decided to drop the Turkey data as of chapter 4.1.3 for the prediction analysis, because missing values cannot be properly handled by the models. A description of the final dataset and its columns is provided in Table 2.

Table 2 Main Dataset columns and their description

Column	**Description**
NUTS-3_code	NUTS-3 region code
year	Time period of the data
NUTS-3_name	NUTS-3 region name
country	NUTS-3 region country
questioncount	Number of questions posted from the region on Stack Overflow
answercount	Number of answers posted from the region on Stack Overflow
commentcount	Number of comments posted from the region on Stack Overflow
upvotecount	Number of upvotes for the region on Stack Overflow
downvotecount	Number of downvotes for the region on Stack Overflow
EMP (THS)	Employment in thousands
GDP (MIO_EUR)	Gross Domestic Product in million euros
GVA (MIO_EUR)	Gross Value Added in million euros
POP (THS)	Population in thousands

3.4 Quantifying Programming Literacy

For all the further modeling and analysis, a singular metric for programming literacy needs to be established. Weighting the different forms of Stack Overflow activities could be a meaningful approach in quantifying programming literacy, as these activities directly reflect the engagement levels of programmers in each region. By weighting these metrics, it is possible to differentiate between passive and active forms of engagement (like voting versus posting or

answering questions), thus identifying regions with not just high traffic but also high participation and expertise in programming.

Questions, answers, and comments on Stack Overflow are tangible indicators of an active programming community. They represent not just the problems and challenges faced by programmers but also their willingness to share knowledge and collaborate on solutions. Posting questions is a primary indicator of active problem-solving and learning in the programming community. It shows a need for knowledge and a willingness to seek assistance, which is fundamental in a vibrant and growing programming hub. A high volume of questions suggests a significant level of engagement in programming activities. Therefore, a weight of **40 percent** is assigned to Stack Overflow questions.

Providing answers is equally crucial, as it demonstrates expertise and a community's capacity to support its members. A region with a high number of answers indicates not only active participation but also a certain level of proficiency and knowledge-sharing culture in the programming community. Thus, answers are given a weight of **40 percent**.

Comments, while important for clarifying questions and answers, are generally less indicative of programming activity than the questions and answers themselves. Thus, comments are given a weight of **10 percent**. Upvotes and downvotes are indicators of community engagement and the quality of content being produced. However, they are more passive forms of participation compared to asking questions, providing answers, or commenting. Therefore, they are assigned a lower weight of **5 percent** each. To summarize, we define programming activity as a weighted summation of questions answered (40%), answers given (40%), number of comments (5%) as well as number of up- and downvotes (5%).

It is important to note that these weights are solely based on a subjective assessment of the importance of the different Stack Overflow activities. Although there is no scientific research that could justify these choices, we think that this methodology is reasonable.

4 Geographical Analysis

In this chapter of the thesis, we delve into a geographical analysis to explore how programming activity is distributed and how it correlates with economic factors. The focus lays on identifying the most important programming hubs and understanding the density of programmers in relation to population and GDP across various regions. This analysis also lays the groundwork for the subsequent chapter, where we will probe deeper into predictive variables for high programming activity in urban areas. Understanding these predictors is crucial for identifying the characteristics that make a region conducive to a thriving programming community.

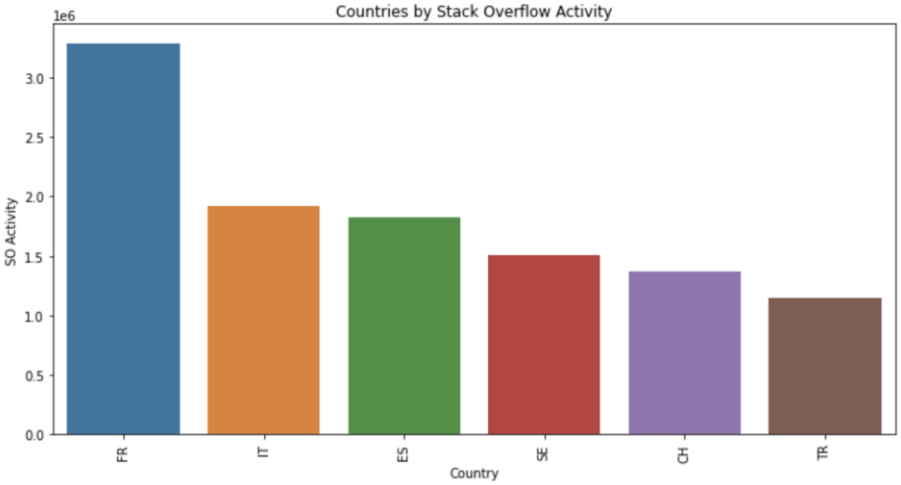
4.1 Identification of Programming Hubs

To identify the most important programming hubs by NUTS-3 regions it is crucial to establish certain criteria that define what constitutes as *important* in this context. Given the information that is included in our dataset, there are several approaches and steps to consider. First, the weighted programming activity is investigated on both country- and region-level. Then, the growth of this activity over time is analyzed to see if it influences the results.

4.1.1 Country-Level Analysis

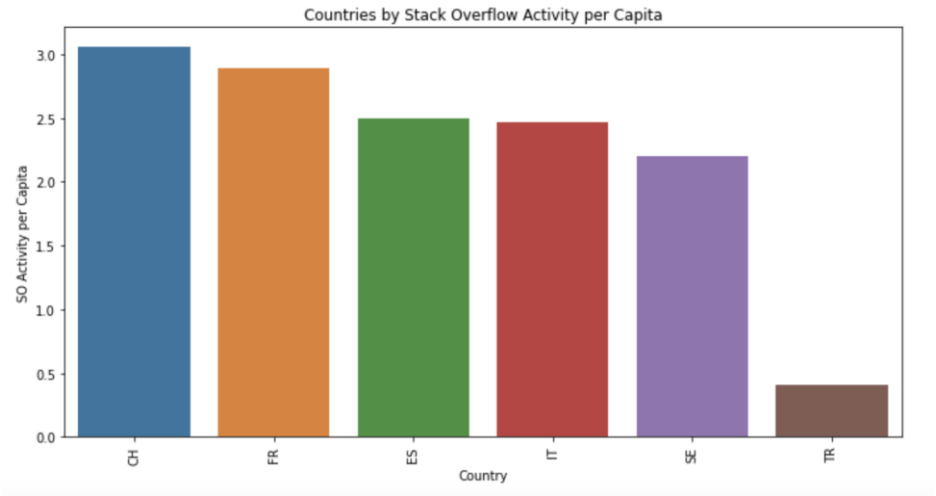
We start by investigating how the programming activity differs on a country level. When plotting the activity and activity per capita for the six countries, significant differences in their ranking can be observed.

Figure 4 Amount of Stack Overflow's Activity across the analyzed countries.



In Figure 4, France (FR) appears to be the leading country in terms of total programming activity, reaching more than three million. Italy (IT) has the next highest level of activity, followed by Spain (ES), Sweden (SE) and Switzerland (CH). All four have lower activity levels than France but are quite close to each other. Turkey (TR) has the least activity among the presented countries, with a value just under one and a half million.

Figure 5 Size of Stack Overflow's Activity per Capita across analyzed countries.



In Figure 5 the y-axis represents Stack Overflow activity per capita, adjusting the activity levels in proportion to the population size of each country, which is a more accurate depiction of the programming literacy across NUTS-3 regions. Switzerland now appears to have the highest level of activity per capita, which suggests that while the overall activity in Switzerland might

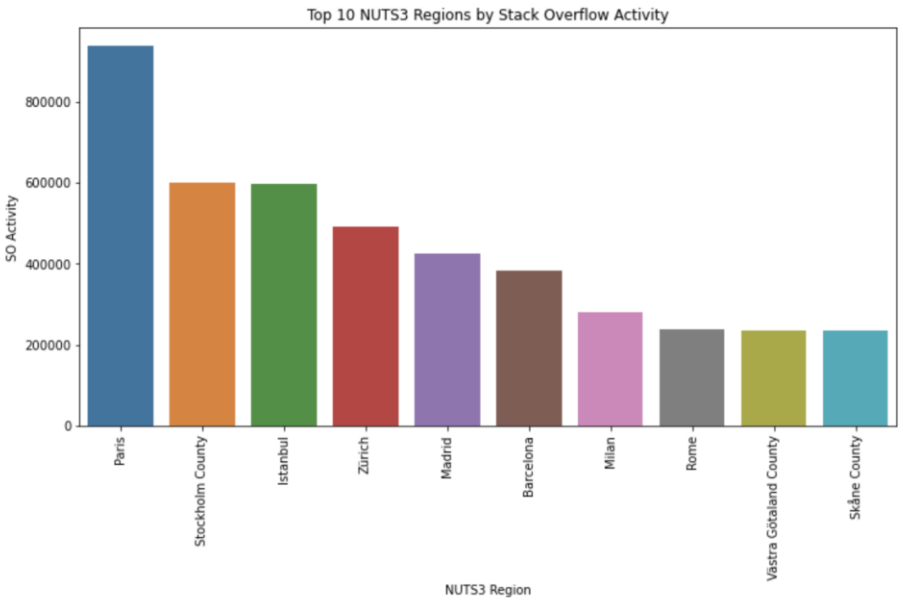
be lower, the engagement per person is higher compared to other countries. France has the second highest score, thus performing very well in both total activity and activity per capita. Spain, Italy, and Sweden still show significant activity per capita. Turkey not only remains the country with the lowest activity per capita, but also shows a huge discrepancy towards the other countries with less than 20% of Sweden, which has the second lowest activity per capita.

Comparing the two Figures, it's evident that this adjustment provides a more nuanced understanding of Stack Overflow activity, suggesting that while some countries may not have the highest overall numbers, their smaller populations are very active on the platform.

4.1.2 NUTS-3-Level Analysis

When performing the same analysis on a NUTS-3 level, there are similar differences that can be observed.

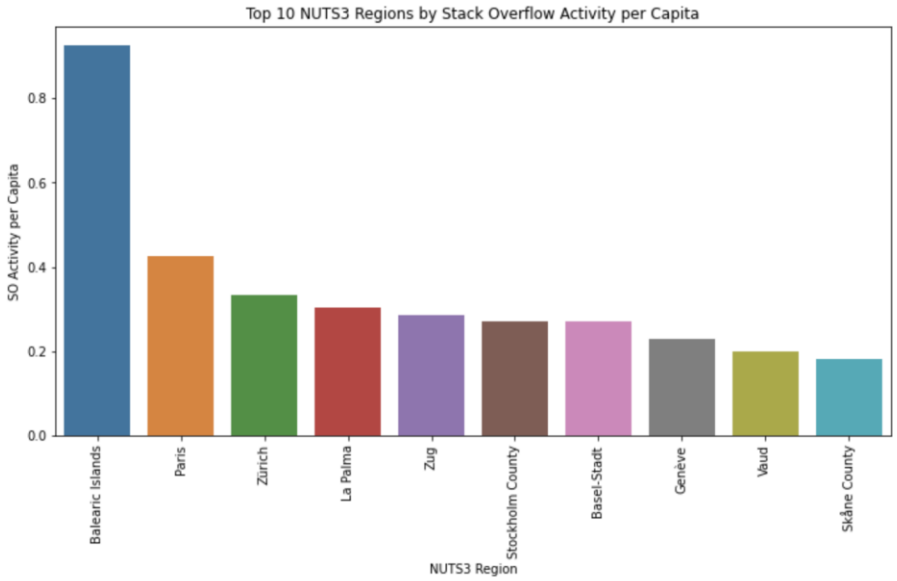
Figure 6 Top 10 NUTS-3 Regions based on the Stack Overflow's Activity Size



With Figure 6 visualizing total programming activity in absolute numbers, we can observe that Paris has the highest Stack Overflow activity, followed by Stockholm County and Istanbul. Except for Switzerland, all of the capitals of the examined countries are represented. This could

be due to various factors such as a larger population of programmers, a strong tech industry, or a combination of both.

Figure 7 Top 10 NUTS-3 Regions based on the Stack Overflow's Activity Size per Capita



The second graph (Figure 7) changes the perspective again by showing programming activity per capita. This measurement gives us an insight into how active the programming community is relative to the size of the population in each NUTS-3-region. Interestingly, the Balearic Islands lead in per capita activity, despite not being the top in overall activity, indicating a highly active programming community relative to its population size. Paris, while still high in per capita activity, is not as dominant as it was in the total activity graph. With Zürich, Zug, Basel-Stadt, Genève and Vaud, Switzerland seems to have many hubs with high engagement per capita. The average programming activity per capita across all regions is 0.035 which highlights the dimension of activity in the displayed programming hubs.

Major cities like Paris, Zurich, and Stockholm County show high activity in both total and per capita terms, indicating both a high number of users and a high engagement relative to population. The presence of smaller regions such as Zug and Basel-Stadt near the top of the per capita graph suggests that smaller populations can have a very active core of Stack Overflow

users. Differences in the rankings between the two graphs highlight the impact of population size on Stack Overflow activity measurements. Some regions may seem less active overall but are very active on a per capita basis.

Overall, the activity per capita graph is crucial for understanding the true level of engagement within smaller populations, while the total activity graph is more reflective of the raw number of interactions on Stack Overflow, which can be influenced by larger populations.

4.1.3 Key Findings - Growth Over Time

So far, only the programming activity in the overall time period from 2008 until 2020 has been investigated. Another approach may be to analyze the growth of activity over time. This is crucial for identifying emerging hubs and understanding the dynamics of programming engagement in different regions. A region experiencing a rapid increase in Stack Overflow activities might indicate a surging interest in programming or a growing IT sector. This time-based analysis would help to distinguish established hubs from fast-growing, emerging ones. The results can be valuable for companies and educators to understand where the tech community is growing and to target these areas for investments, recruitment, or expansion of educational programs.

The Compound Annual Growth Rate (CAGR) is computed as a metric for programming activity over time. It measures average annual growth over a given period.

$$CAGR = \left(\frac{\text{Ending Value}}{\text{Beginning Value}} \right)^{\frac{1}{\text{Number of Years}}} - 1$$

When looking at Figure 8 and Figure 9 can observe that seven out of the ten regions displayed in both graphs are in Italy. All these regions have an average annual growth rate above 80%, which is twice as high as the average growth rate per region of 39%. This indicates that Italian

regions experience a significant increase in Stack Overflow activity over time. When computing the CAGR and the CAGR per capita on the country level, Italy shows the second highest overall growth (41%) just after France (44%) which confirms this indication.

Figure 8 Top 10 NUTS-3 Regions based on the GDP Growth defined as CAGR

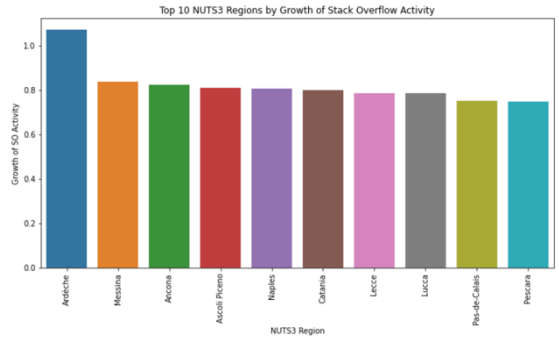
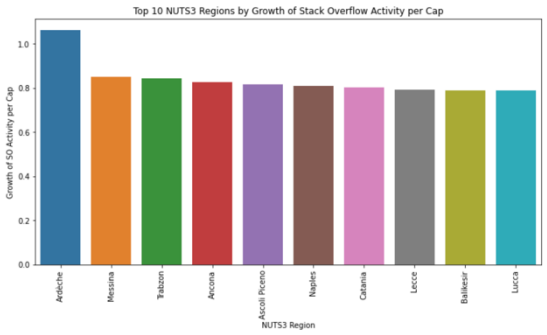


Figure 9 Top 10 NUTS-3 Regions based on the GDP Growth defined as CAGR per Capita



The order of regions by growth rate appears quite similar in both graphs, suggesting that the growth in activity correlates with growth in activity per capita. Regions like Ardèche, Messina and Trabzon lead in both graphs, highlighting that they are not just becoming more active in absolute terms but are also seeing significant engagement growth relative to their population. The presence of the same regions in both graphs suggests that the factors driving the growth of Stack Overflow activity in these areas are affecting the population uniformly, rather than being concentrated in a subset of the population (such as might be the case if growth were driven by migration of tech workers into the region).

What is interesting is that none of the NUTS-3 regions that showed high total programming activity and high activity per capita are represented in these graphs. This could be due to the fact that more established programming hubs don't experience the same level of growth as smaller emerging hubs. All in all, programming activity in absolute terms offers a raw count of how much coding is happening in an area, pointing to regions with the most vibrant tech ecosystems. Per capita Figures adjust this activity for population size, identifying places where programming is not just common, but a central part of the environment. On the other hand, the

growth of programming activity merely shows the increase or decrease over time, which doesn't necessarily reflect the current size or density of programming hubs, making it a less immediate indicator of where the most active communities are.

4.2 Programming Density based on the "Programmer" Definition

Building on the prior chapter's identification of principal programming hubs within Europe, this section delves into a granular analysis of programming density and its influence on Gross Domestic Product (GDP) across regional landscapes. Here, we define programming literacy as the breadth of programmer presence, gauged by Stack Overflow activities. Hence, 'programmers' are those who demonstrate consistent engagement with the platform, indicative of programming literacy.

We derived our definition of a 'programmer' from the analysis of Stack Overflow user activities, such as posting questions, providing answers, and engaging with the content through votes and comments. This data, collated by year and NUTS-3 region, is enriched by insights from the Stack Overflow Developers Survey 2021, which offers detailed accounts of interaction frequency and professional behaviors. This dual-faceted approach enables a nuanced understanding of the term 'programmer.'

Employing clustering techniques, specifically K-modes clustering, we analyzed the 2021 survey data to identify distinct patterns of engagement among users. K-modes, an algorithm particularly adept at handling categorical data, sorts users into clusters based on their activity patterns and professional attributes (Chaturvedi et al. 2001). This method's strength lies in its ability to provide clear interpretability of complex, categorical datasets. Through this process, we are able to delineate a multifaceted definition of a programmer, capturing the essence of programming literacy as represented in diverse user interactions on Stack Overflow.

4.2.1 Definition of a Programmer based on Stack Overflow Activity Data and Stack Overflow Developers Survey

The initial analysis using K-modes clustering unveils three clusters among 36 combinations of *Profession* and *Frequency*. K-modes clustering identified clusters, representing personas based on respondents' professional behaviors and Stack Overflow participation frequency. Based on these clusters, it is possible to define the following personas:

1. Cluster 0: Comprising over 50% of respondents, Cluster 0 signifies the most common behavior – individuals in the early stages of learning programming or pursuing it as a hobby. These users, not yet professionals, interact either daily or monthly, likely reflecting different proficiency levels.
2. Cluster 1: Encompassing around 25% of respondents, Cluster 1 comprises users who write code occasionally as part of their work. Less active on Stack Overflow, they engage in discussions on a monthly basis or even less frequently.
3. Cluster 2: The smallest cluster, Cluster 2, likely represents professionals working daily as programmers. Their frequent engagement in Stack Overflow discussions, a few times per week, marks them as highly involved platform users.

In summary, the clustering results offer valuable segmentation, laying the groundwork for more detailed analysis and targeted strategies based on identified personas. Each of the defined personas are described in the Table below (Table 3). They are segregated based on the clustering results (modeled on the Profession and Frequency variables).

To estimate the number of programmers in NUTS-3 regions, specific activities in the main dataset are considered for each persona. The variables *answercount, questioncount, upvotecount, downvotecount,* and *commentcount* are used to identify and quantify each persona's engagement, with adjustments made for the frequency of their participation.

Table 3 Defined Personas based on the Clustering Modelling of Stack Overflow Survey Answers

Persona	**Expert**	**Enthusiast**	**Late Adopters**
Profession	Programmer	Not a programmer, but can code	Student, learning how to code
Frequency of Participating in SO	Few times per week	Once per month	Almost daily
Type of Interaction	Answers	Questions	Reactions (votes, comments)

New variables are created to estimate the number of total programmers in the region. Each type of interaction is assigned to a specific persona, as presented in Table 3 and then adjusted based on the frequency they present to calculate the final number of Experts, Enthusiasts and Late Adopters in the NUTS-3 region. Then, the total number of programmers is calculated per region and year as a sum of all personas. This comprehensive approach allows for granular interpretation and application of these personas to real-world data. Interestingly, some users are not professional programmers, supporting our definition that programming literacy doesn't have to be tied to being a professional programmer.

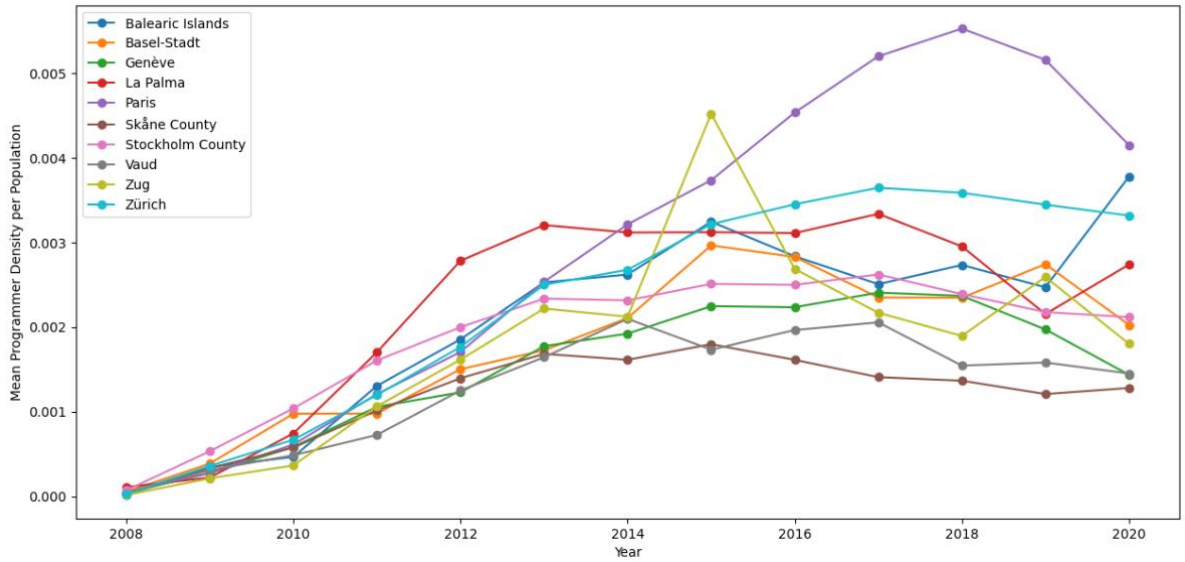
4.2.2 NUTS-3-level Programmers Density Analysis

Based on the established definition on previous stage, the total number of programmers was calculated. It emerges as a significant factor examined in this chapter within the context of key economic indicators such as GDP, Employment, and Population. The objective of this chapter is to enhance comprehension of the correlation between the estimated number of programmers and regional economic development.

The first analysis in Figure 10 reveals that Paris emerges as the leader with the highest estimated programmers count, followed by Stockholm County, Zurich, Madrid and Barcelona, indicating a high level of programming literacy. In 2020 the Balearic Islands stand out as one of the cities with the highest number of programmers per capita, after Paris. This defies expectations for a

region typically recognized for tourism rather than technology (Pons and Rullan 2014). Notably, other high-density locations align with Europe's technological hubs (Paris, Stockholm), attracting programmers, most probably due to the presence of tech companies and well-developed business environment. This proves that most programmers tend to live and work in big technological hubs, however presence of regions like Balearic Islands and La Palma and their rapid growth in 2019-2020, should indicate the existence and growth of the remote work trend across programmers in Europe.

Figure 10 Programmers Density by Population by NUTS-3 Region from 2008-2020



Summarizing findings from Figure 10, while the number of programmers grew over time and therefore programming literacy, the growth concentrated within established tech hubs, indicating their sustained attractiveness to this demographic.

Further analysis of Programmers Density per GDP, presented on Figure 11, has shown that Balearic Islands boast the highest programmer density per GDP. Later analysis has shown that Balearic Islands and La Palma demonstrate high programmer density despite lower GDP per capita, suggesting inefficiency of tech sectors or lack of local companies leading local growth. Another reason could be that their popularity for remote work resulted in the emergence of a new industry, accounting for a considerable proportion of total GDP. Paris and Zurich together

with other NUTS-3 regions have significantly lower density of programmers per GDP, which can be explained by their diverse economies, showcasing their economic strength through various industries. Noteworthy deviations, such as in Spain (Balearic Islands and La Palma), could be traced back to the impact of remote work trends and housing dynamics on programmer distribution (Benitez and Castillo 2023). However, the nature of the data, which presents Stack Overflow activity makes it difficult to understand whether identified programmers are long term habitants of those regions or not, determining their status as digital nomads.

Figure 11 Most Programmer Dense per GDP NUTS-3 Regions across the Years

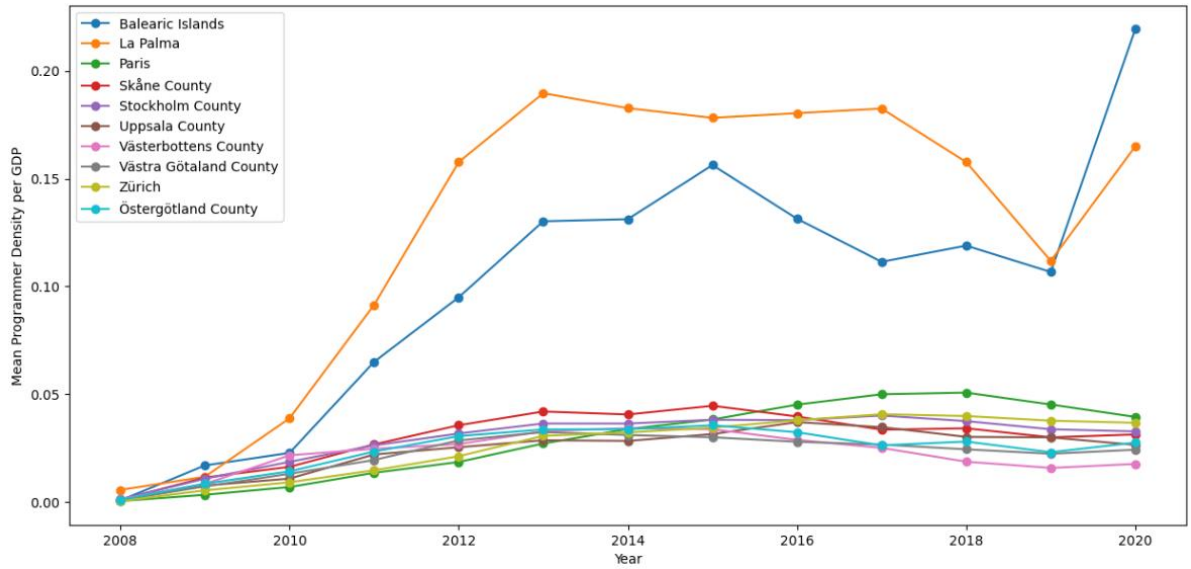
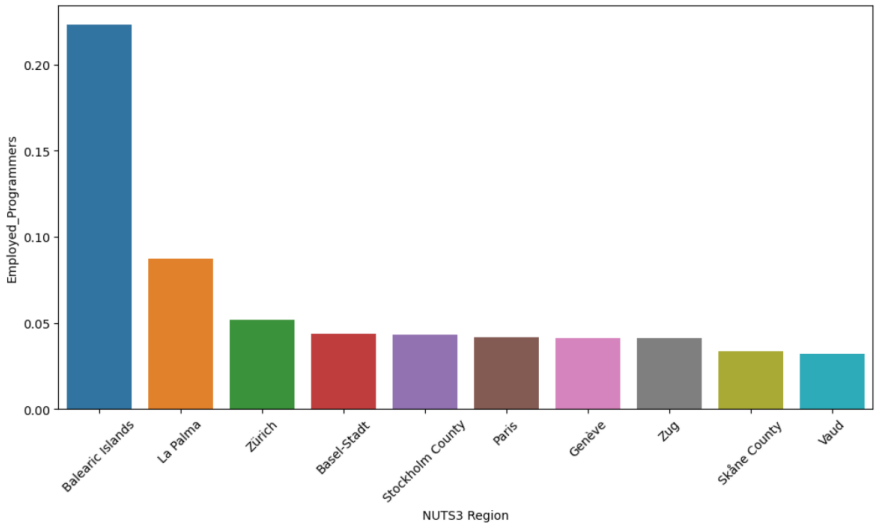


Figure 12 presents the ratio of programmers to the total employed population per region and provides additional insights. Balearic Islands and La Palma continue to showcase a strong tech industry presence. Major economic hubs like Zürich, Basel-Stadt, Stockholm County, Paris, Genève, and Zug exhibit noteworthy ratios, indicating a significant tech workforce contributing to the overall employment landscape. These regions often serve as economic centers with diverse industries. Paris has a high density of programmers per population (Figure 10), but a lower density per employment suggests several conclusions. As a major global city, it likely has a diverse economic landscape with employment opportunities in various sectors. While it attracts programmers, they may not represent the majority of the employed workforce. Other

industries and professions might contribute more significantly to the overall employment makeup. Also, Paris has a much higher employment rate than the rest of the regions analyzed (89% employment rate, with dataset average of 53%), which affect the ratio of programmers when compared to total workforce of Paris, the most employed region across analyzed NUTS-3.

Figure 12 Ratio of Programmers per Employment based on the top NUTS-3 Region



In conclusion, this comprehensive analysis uncovers the dynamics of programming literacy (defined as the number of programmers) and economic development in terms of GDP, employment rates and population size. The cities that observe high programmers' density per GDP (Figure 11), tend to also have higher ratio of programmers in all employed population (Figure 12). This observation suggests that cities with a higher literacy rate are often characterized by more tech-driven economies. These tech-driven economies tend to attract talent in the field of programming, thereby reinforcing this trend and creating a positive feedback loop. They also tend to invest in infrastructure and education. The latter might explain Paris' higher literacy to GDP proportions in comparison to employment, as it is a popular destination for tech-savvy individuals enrolled in quality educational opportunities.

As mentioned previously, disparities in programmer distribution between regions such as the Balearic Islands and La Palma may be significantly influenced by evolving trends in remote work and local housing dynamics, especially since the pandemic. These factors potentially affect the concentration of programmers in various areas, especially in relation to urban centers. This phenomenon presents an intriguing case study, suggesting that these regions could serve as valuable outliers for more in-depth analysis to understand how remote work preferences and residential choices are reshaping the geographic distribution of technology professionals.

5 Predicting Programming Literacy

This chapter is dedicated to establishing the predictive power of economic variables over programming literacy at the NUTS-3 regional level. Informed by our preceding analysis, we posit a correlation between regional economic health and programming literacy, defined here as engagement in Stack Overflow discussions. Such literacy serves as the dependent variable in our predictive models, casting the efficacy of economic indicators as a barometer for forecasting programming engagement.

Tackling the prediction of programmer density within urban areas, we build on the comprehensive preprocessing and methodological foundation set in earlier sections. Four predictive models have been crafted and critically assessed: Ordinary Least Squares (OLS), Ridge, Lasso, and Random Forest. Our mission is to calibrate these models to not only forecast programming activities with high precision but also to ensure an optimal balance between accuracy and generalizability.

5.1 Feature Selection

To develop a strong basis for our models, the process is initiated by finding prospective predictors that have significant correlation with programming density. Correlation analysis

against programming activity indicates that economic indicators such as GDP, employment rates, and Gross Value Added (GVA), as well as demographic variables including population dynamics may be valuable predictors for the model. As a starting point an Ordinary Least Squares (OLS) model is chosen as the baseline for our cross-sectional 2016 analysis. OLS (Ordinary Least Squares) is a statistical method used to find the straight line in linear regression by minimizing the sum of squared differences between observed and predicted values (Lewis-Beck 1980, 9-12) and therefore is most suitable for predicting values that share linear relation.

To determine the most effective set of predictors, we conducted a comparative analysis using various models. The initial model used GDP (in millions of euros) as a standalone measure of economic productivity. Subsequently, we disaggregated GDP into two distinct variables: GDP per capita and Population. This approach allowed us to scrutinize their individual contributions to the predictive model. Since GDP per capita is derived from the total GDP and population size, it was imperative to assess the interplay of these factors within our model. We utilized coefficient and p-value analysis to gauge the impact of each predictor, with coefficients indicating the strength and direction of the variable's effect on programming literacy, as delineated in Tables 4 and 5.

Table 4 Independent Variables Statistics based on the OLS Regression (without GDP per capita)

	coef	**std err**	**P>|z|**
const	-385.8516	653.625	0.555
EMP (THS)	16.4822	26.450	0.533
GDP (MIO_EUR)	0.2331	1.516	0.878
GVA (MIO_EUR)	0.1021	1.608	0.949
POP (THS)	-10.8342	9.807	0.269

GDP (MIO_EUR) has a positive coefficient of 0.2331, suggesting that as GDP increases, programming literacy also increases. POP (Population) has a negative coefficient of -10.8342,

indicating that as the population increases, the dependent variable decreases. The statistical significance (P>|z|) values associated with each coefficient in the models help assess whether the estimated coefficients are significantly different from zero. A lower P-value suggests higher statistical significance.

- For GDP (MIO_EUR), the P>|z| value is 0.878, which is relatively high. This suggests that the coefficient for GDP (MIO_EUR) is not statistically significant at conventional significance levels (e.g., 0.05), indicating that the effect of GDP on the dependent variable may not be reliably different from zero.
- For POP (Population), the P>|z| value is 0.269, also relatively high. This suggests that the coefficient for Population is not statistically significant as well, indicating that the effect of population on the dependent variable may not be reliably different from zero.

Table 5 Independent Variables Statistics based on the OLS Regression (with GDP per capita)

	coef	**std err**	**P>|z|**
const	1209.1301	1427.264	0.397
EMP (THS)	14.7173	26.085	0.573
GDP (MIO_EUR)	-0.0130	1.510	0.993
GVA (MIO_EUR)	0.4254	1.618	0.793
GDP_per_cap	-0.0523	0.039	0.185
POP (THS)	-11.5291	9.839	0.241

The second model consists of an additional variable GDP per Capita (GDP_per_cap). GDP_per_cap has a negative coefficient of -0.0523, indicating that as GDP per capita increases, programming literacy decreases. GDP (MIO_EUR) has a very small negative coefficient of -0.0130, suggesting a minimal effect on the dependent variable. POP (Population) has a negative coefficient of -11.5291, similar to the first model. Comparing p-values it can be concluded that:

- For GDP (MIO_EUR), the P>|z| value is 0.993, which is very high. This further confirms that the coefficient for GDP (MIO_EUR) is not statistically significant, supporting the notion that breaking GDP into components may provide more meaningful insights.
- For GDP_per_cap, the P>|z| value is 0.185. While this is lower than the P-value for GDP in the second model, it is still relatively high, suggesting caution in interpreting the statistical significance of the GDP_per_cap coefficient.
- For POP (Population), the P>|z| value is 0.241, which is again relatively high, however is smaller than in the first performed model.

Comparing the two models, it seems that GDP (MIO_EUR) in the first model has a positive impact, while in the second model, GDP per capita has a negative impact. However, the coefficient for GDP (MIO_EUR) in the second model is very small, suggesting that breaking GDP into GDP per capita and population might be more informative.

The baseline OLS model that includes GDP per capita performs well also across other statistics. The Gross Domestic Product was not included in final model, because as stated previously it is not statistically significant for prediction of programming literacy. The results from the second attempt prove the initial assumptions (refer to Table 6 and Table 7).

Table 6 Independent Variables Statistics based on the OLS Regression (with GDP per capita)

	coef	**std err**	**P>|z|**
const	1201.6333	1514.344	0.427
EMP (THS)	14.6540	24.597	0.551
GVA (MIO_EUR)	0.4113	0.156	0.008
GDP_per_cap	-0.0521	0.037	0.154
POP (THS)	-11.5108	10.069	0.253

The second attempt to define the OLS model reveals that approximately 86.8% of the variability in programming activity is explained by economic factors, as indicated by the adjusted R-squared value. The model results in an F-statistic of 29.68, and the associated probability (Prob (F-statistic)) is very low (8.52e-21).

Table 7 Results of the OLS (Ordinary Least Squares) Regression Model (with GDP per capita)

Statistics:	**Value:**
R-squared:	0.868
Adj. R-squared:	0.866
F-statistic:	29.68
Prob (F-statistic):	8.52e-21

This low p-value suggests that the overall model is statistically significant, and the regression coefficients are not all equal to zero, although a high condition number suggests potential multicollinearity issues. This condition complicates the interpretation of individual predictor effects and can destabilize the regression coefficients, although the final set of independent variables can be defined. For the purpose of this study, statistically significant variables are Employment, Population, Gross Value Added and Gross Domestic Product per Capita.

5.2 Model Selection

To address potential multicollinearity and overfitting issues identified in the OLS model, Ridge and Lasso regressions are employed using independent variables including GDP per Capita. Recognizing the need for a robust evaluation metric, cross-validation scores are chosen to assess the performance of each model. Cross-validation involves partitioning the data into subsets, training the model on one subset, and evaluating it on another, providing a more reliable estimate of generalization performance. Several models are tested to compare performance across cross validation scores, as presented in Table 8.

Ridge and Lasso regressions are models that efficiently address multicollinearity issues, providing a balanced approach to controlling overfitting and identifying key predictors. Both models perform well, with Lasso slightly ahead in cross-validation scores due to its feature selection capability. Ridge does not improve the score achieved for Linear Regression. The Random Forest Regressor is investigated to capture any non-linear relationships. Although it shows a high cross validation score of 0.714 on the training set, the discrepancy between the training dataset cross-validation scores of 0.943 indicates potential overfitting. Based on the comparison of cross validation scores, Lasso regression is chosen as a final predicting model.

Table 8 Comparison of performance of multiple predictive models

Predictive Model	**Cross Validation Score**
Linear Regression	0.7235
Lasso Regression	0.7238
Ridge Regression	0.7235
Random Forest Model	0.7140

As a next step, the growth rate from the last 5 years is calculated and added as an additional variable, in order to calibrate the efficiency of the model. The reason behind this is to learn to model, predict and adjust to long term trends. The best performing model, Lasso Regression, was used to predict programming literacy from 2017-2020 based on the. Results from the Lasso Regression model are presented below in Table 9.

Table 9 Summary of the Lasso Regression Model with growth rate

Statistics:	**Value:**
Mean Cross-Validation Score:	0.650
Standard Deviation of Cross-Validation Score:	0.225
F-statistic:	272,02

5.3 Interpretation of the Results

In summary, the integration of economic indicators with various modeling techniques establishes a robust framework for predicting programming literacy, a crucial factor in guiding strategic urban technological development. Among these techniques, lasso regression emerges as the best-performing method, offering insights into the predictive power of economic factors on programming proficiency. The correlation between the predicted and observed values based on the cross-validation score of 0.65 across all the years validates the model's effectiveness and indicates that economic growth factors can be used for forecasting programming literacy in diverse geographical regions. Overall, the model shows high performance, indicated by consistently high F-statistics scores across all tests, reflecting a statistical strength of predictions.

However, the cross-validation score has not improved much in comparison to previous tested models. To investigate this issue, a brief examination of outliers is warranted. It is important to note that a more thorough analysis and discussion of outliers will be presented in the next chapter. While the economic factors explored demonstrate an ability to explain and forecast programming literacy, critical observation emerges. There is a tendency to overestimate programming literacy in regions with lower GDP per Capita/GVA and, conversely, to underestimate in areas where GDP per Capita/GVA is exceptionally high compared to other locations. This suggests a sensitivity to extreme economic conditions, leading to inaccuracies in predictions. Due to this limitation, regions such as the Balearic Islands, where the model overestimates programming literacy despite lower GDP per Capita, and locations like Paris, where the model underestimates activity due to exceptionally high GDP per Capita, highlight the limitations of relying solely on economic factors (refer to the Table 10). This prompts the realization that programming literacy is influenced by a broader spectrum of variables beyond economic factors, potentially encompassing social dynamics and local business characteristics.

Even though a model can predict programming literacy based solely on economic factors, to enhance the model's stability and reliability, a comprehensive approach should involve integrating indicators from education, social conditions, and industry-specific data, ensuring a more comprehensive and reliable source of information for predicting programming literacy.

Table 10 Analysis of chosen regions based on predicting model of programming literacy

NUTS-3 Name	GDP	GVA	Employment	Activity*	Residual
Balearic Islands	3’106.9	2’821.2	52.9	6’648.4	3’495.3
Paris	234’421.0	208’514.2	2’037.5	114’794.0	-27’229.0

* Variable “activity” is a target variable interpreted as “programming literacy”

5.4 Outliers

5.4.1 Outlier Identification

In predictive modeling, outliers can significantly influence the results and interpretations. In this research, the presence of outliers, especially in large towns with distinctive economic profiles, necessitates a focused analysis. These outliers are not just statistical anomalies; they often represent unique socio-economic environments that can provide valuable insights into the dynamics of programming literacy.

Outliers are detected and identified based on the residual analysis. With residual analysis, it is possible to analyze the differences between the predicted programming literacy and the actual programming literacy recorded in the NUTS-3 region. The results are easy and intuitive to analyze, the higher the difference, either negative or positive, the more outlined the prediction is (Agresti 2002). Our identification of outliers is further supported by the visual analysis of the residuals from our predictive model, as depicted in Appendix 1. The chart displays a distribution with a few extreme values that are quite distant from the rest of the points. Such visual evidence underscores the need to address these outliers comprehensively, as they could

represent regions with anomalous economic growth or programming density that do not align with the general trends.

Based on the residuals analysis, a few major outliers can be defined. To identify what constitutes an outlier, the standard deviation is calculated. Based on that value the regions that record residuals higher than the standard deviation are considered outliers.

Table 11 presents the most deviated outliers from the prediction. The first three rows correspond to the most overestimated regions in terms of programming literacy, while the bottom three rows represent the most underestimated regions in terms of programming literacy. The analysis identifies outliers as particularly large towns with high GDP per Capita and employment rates. These include major cities like Paris, Madrid, Stockholm, and Zurich, which are primarily capital cities and major economic hubs. These cities stand out due to their unique patterns in programming literacy, which deviate from the trends observed in other regions.

Table 11 Outliers based on the Residuals analysis made on the base of OLS Regression

Region Name	**Country**	**EMP (THS)**	**GDP per Capita**	**Activity***	**Prediction**
Rome	Italy	2171	38176	24875	More
Hauts-de-Seine	France	1148	108940	24309	More
Naples	Italy	993	19842	6992	More
Paris	France	2037	107931	114794	Fewer
Zürich	Switzerland	864	90147	55345	Fewer
Stockholm County	Sweden	1319	64446	58633	Fewer

* Variable “activity” is a target variable interpreted as “programming literacy”

The regions where the model overestimates programming activity share some commonalities. Notably, these regions, such as Hauts-de-Seine in France and Barcelona in Spain, often have relatively high values in both Employment (EMP) and Gross Domestic Product per Capita (GDP_per_cap). This suggests that the model might be influenced by regions with higher

economic and employment activities, potentially leading to overestimation of programming activity. On the other hand, regions where the model underestimates activity, like Paris and Zürich, exhibit significant programming activity relative to their economic indicators. This could indicate that the model may not be capturing the nuanced relationship between economic indicators and programming literacy in these specific regions, resulting in underestimation. The differences between overestimated and underestimated regions highlight the importance of considering regional specifics and potentially refining the model to better capture the factors influencing programming activity in diverse geographical contexts. Further investigation into the specific features contributing to these discrepancies and potential interactions between variables could provide insights for model improvement.

5.4.2 Outliers Analysis

The findings presented in Figure 13 and 14, that neither GDP per Capita nor Employment is correlated with the level of residuals in the model, suggests that these significant variables may not be directly influencing the discrepancies between predicted and actual values. In other words, variations in GDP and Employment do not seem to explain the differences in the model's predictions and the observed programming activity levels.

Figure 13 GDP per Capita vs Residuals per NUTS-3 Regions based on the Outliers

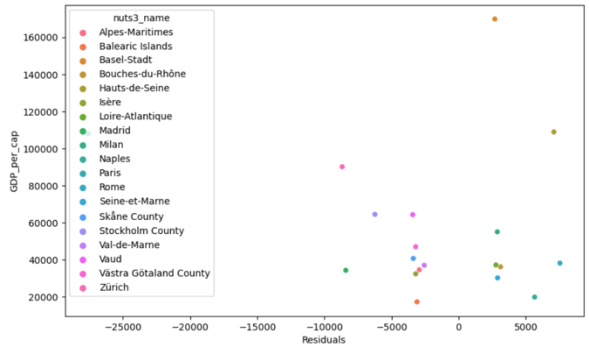
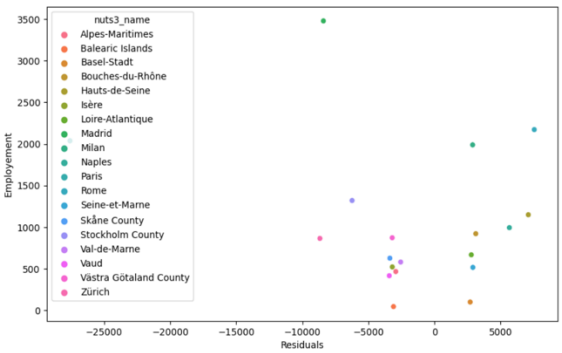


Figure 14 Employment vs Residuals per NUTS-3 Regions based on the Outliers



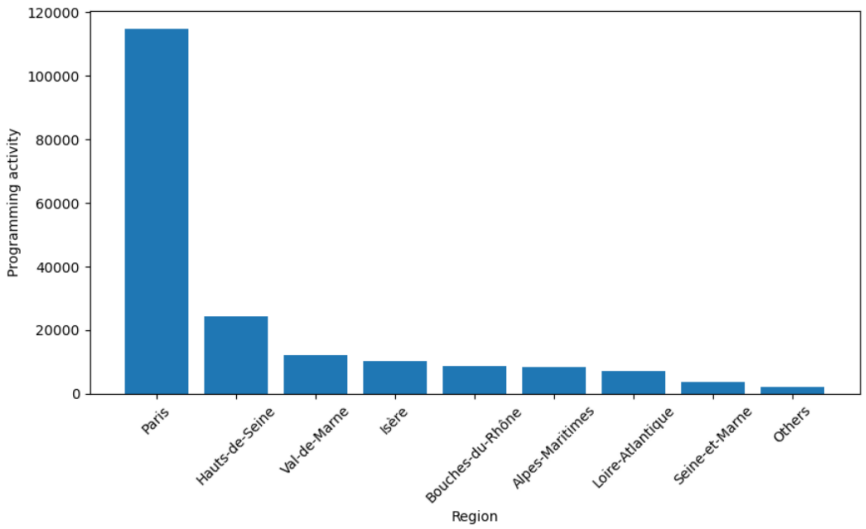
Since GDP per Capita and Employment do not seem to explain the level of residuals, relying solely on these variables for outlier analysis may provide limited insight. Outliers might be

driven by factors not captured by the chosen predictors. Further exploratory analysis is warranted to identify the specific characteristics or regional factors contributing to outliers. This could involve investigating the nature of the residuals, examining potential interactions between variables, and considering region-specific features.

Further investigation shows that France has twice as many regions identified as outliers compared to other countries, with 8 outliers, while the rest of the countries analyzed have an average of 3 outliers. In most cases, top cities such as Barcelona and Madrid in Spain, and Milan, Rome, and Florence in Italy, are identified as outliers. Surprisingly, Switzerland has only 3 outliers, even though the GDP per Capita of the regions in that country is significantly higher than in other countries with a greater number of outliers.

To understand the dominance of France in terms of outliers, a more in-depth analysis is needed. Figure 15 below, shows that Paris dominates in programming activity, exhibiting significantly higher values than other regions, thus indicating a strong presence in programming literacy.

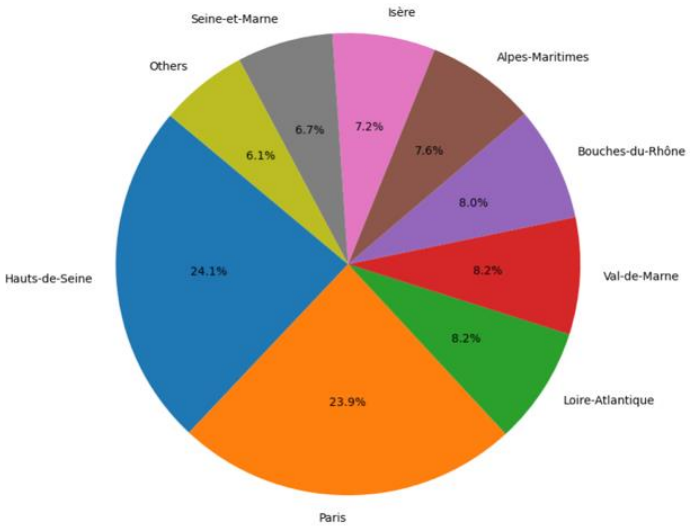
Figure 15 Distribution of Average Programming Literacy across the outlier regions and rest of the French NUTS-3 Regions.



There are substantial differences in programming literacy activity between Paris and other regions, suggesting regional disparities in programming literacy initiatives. The 'Others' category represents the combined programming literacy (defined as “activity”) of multiple

regions, contributing less individually but significantly in aggregate. It shows that all regions that are identified as outliers have higher programming literacy than the remaining NUTS-3 regions identified properly. This proves that the model is not adjusting well to higher programming literacy levels.

Figure 16 Distribution of Average GDP per Capita across the outlier regions and rest of the French NUTS-3 Regions.



As a next step, the GDP per capita distribution is analyzed as shown in Figure 16. Paris and Hauts-de-Seine stand out with high GDP per capita compared to other regions, as well as Loire-Atlantique and Val-de-Marne. The 'Others' category, consisting of the remaining NUTS-3 regions, presents lower mean GDP per capita than the outlier regions. This suggests that the discrepancy in mean GDP per capita in outlier's regions might influence the model's suboptimal predictions, a hypothesis that could be validated by scrutinizing individual locations one by one. France's top region shows the same behavior. Renowned for being one of the wealthiest regions in France, Hauts-de-Seine is a major hub for European commerce. One of Europe's principal commercial hubs, Paris, is primarily responsible for the region's economic prosperity. Examining French administrative departments reveals diverse characteristics. Economic significance in departments like 'Hauts-de-Seine,' near Paris, may be due to major business districts, while 'Alpes-Maritimes' contribute primarily through tourism and services. Variations

in programming literacy and GDP per capita across French regions result from factors such as geographical remoteness, economic focus, and proximity to urban centers. France could be considered an outlier due to its higher-than-average GDP per capita, employment, and economic activity levels compared to the other countries in the dataset. The large standard deviations in these measures also contribute to the perception of France as an outlier, suggesting that there is considerable regional variation within the country, which model could not capture well, without additional dimensions.

In summary, outliers with higher actual programming literacy are predicted with lower, and vice versa, which seems to be an interesting finding. This is probably caused by the lack of additional dimensions; hence the model cannot grasp the more complicated relations between programming literacy and economic growth. Therefore, it is generalizing all regions and averaging the programming literacy across the whole dataset.

5.4.3 ARIMA and non-stationarity in the outliers' regions

Building upon the initial analysis of outliers in predictive modeling, this part progresses towards a nuanced exploration of these regions using ARIMA (AutoRegressive Integrated Moving Average) modeling. This approach focuses particularly on the non-stationary aspects of the identified outlier regions. Non-stationarity, characterized by statistical properties of a time series that change over time, poses a significant challenge to model predictability and stability. It is particularly relevant when exploring socio-economic environments like those represented by outliers, where standard analysis methods may not capture all influencing factors.

In this context, Rome and the Balearic Islands emerge as regions of interest due to their non-stationary behavior. This suggests that these areas may experience economic and social changes not typically captured by more traditional models. By employing ARIMA modeling, a deeper understanding of the time-based dynamics within these regions is sought, aiming to identify

underlying patterns and trends that affect programming literacy. The choice of a p-value threshold of less than 0.05 in the ARIMA model is critical in this regard. It ensures that only the most statistically significant patterns in the time series data are considered for the analysis. This filtering criterion aids in isolating regions where there are meaningful, non-random fluctuations in data over time.

The identification of non-stationarity in regions like Rome and the Balearic Islands through ARIMA modeling offers valuable insights. It underscores the presence of dynamic economic factors that potentially influence programming literacy in ways that deviate from general trends observed in other regions. Figures 17 and 18 depicting the GDP per capita over time for these regions further illustrate this point. Fluctuations in GDP per capita in both Rome and the Balearic Islands indicate economic variability, which might be influencing the programming literacy environment in these regions.

Figure 17 GDP per Capita Over Time for Balearic Islands

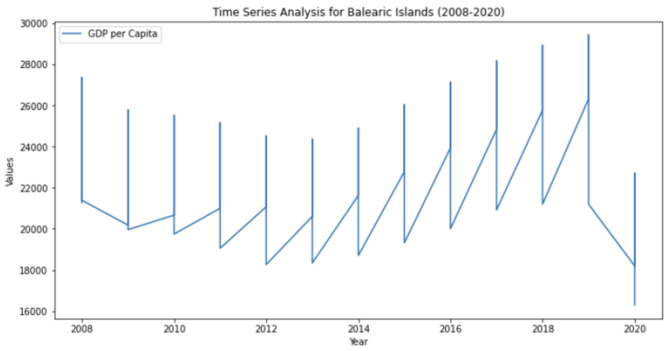
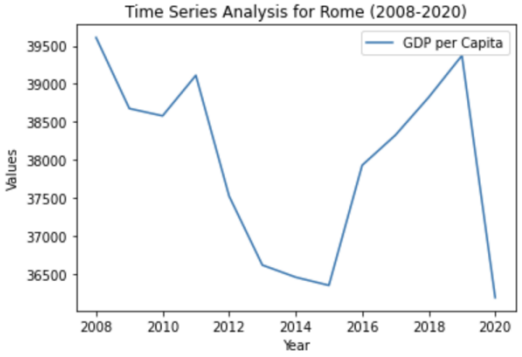


Figure 18 GDP per Capita Over Time for Rome



The ARIMA analysis of non-stationary regions reinforces the complexity of economic patterns and their impact on programming literacy. It suggests that the current model could be augmented to account for fluctuations over time, particularly in dynamic regions like Rome and the Balearic Islands. The findings from this chapter should inform not only the interpretation of the current model's results but also the development of future models.

6 Discussion

In exploring the relationship between programming literacy and economic growth in European regions, the study indeed found a correlation as well as patterns between both dimensions. The study identified major European tech hubs like Paris, Stockholm, and Zurich as central to programming activity, with their status as capital cities or economically dominant regions providing the necessary infrastructure, educational resources, and corporate density essential for thriving programming communities, as theorized in the thesis. Notably, Swiss regions occupied many places in the rankings. An explanation could be Switzerland's strong emphasis on high-quality education, substantial investment in research and development, as well as high living standards attracting global talents, which have collectively fostered a conducive ecosystem for programming excellence and innovation. Unexpectedly, the Balearic Islands and La Palma also demonstrated high programming activity, likely due to their emerging status as remote work havens, while Italian regions, despite initially lagging, have shown significant growth, signaling a broader, policy-influenced tech evolution across Europe.

In addition, the study employed various predictive models to forecast programming literacy using economic indicators. While models like Lasso regression showed potential, the in depth-analysis of outliers revealed limitations, particularly in regions with unique economic profiles or high GDP per Capita. These findings suggest a complex interplay between economic growth and programming literacy, influenced by diverse regional characteristics and evolving trends such as remote work.

As the group study concludes, the forthcoming chapters will delve into a range of focused topics: After conducting panel data analysis, Poland serves as a case study to explore various facets of interest. Subsequently, new variables capturing programming literacy are added and their impact assessed and lastly programming literacy’s effect on economic resilience is investigated.

7 A Panel Data Analysis

7.1 Introduction

This thesis investigates the connection between programming literacy and economic growth, focusing on a dataset that combines Stack Overflow activity with economic factors across NUTS-3 regions. The earlier part of the thesis explored this relationship for the year 2016 using cross-sectional regression, examining how factors like GDP, employment, and GVA relate to programming activity. The model's efficacy was then tested by predicting programming activity, followed by a thorough analysis of outliers.

However, this approach did not fully leverage the panel structure of the data, comprising both temporal and cross-sectional dimensions. In this part of the paper, the aim is to fill this gap by extending the analysis. This involves comparing the performance of different Machine Learning models in explaining programming activity for the whole timeframe of the dataset, rather than just one cross-section. The goal is to see whether the advanced capabilities of ML models can provide new insights or more accurate predictions regarding the residuals, which can be understood as excess programming literacy in this context. The analysis is further enhanced by exploring if these residuals can lead to an improvement in predicting future economic growth, mainly using GDP as an indicator. The findings of this research could help to establish a way to successfully make reliable predictions of economic development in each region or country, based on its engagement in programming activities. This could redefine the value of having such an engagement and thus lead to an extensive change of decision-making in political, economic, and educational institutions.

7.2 Theoretical Fundamentals

7.2.1 Types of Data

Understanding the nuances of cross-sectional, time-series, and panel data is crucial before delving into the topic of panel data regression. Each type presents unique characteristics and analytical challenges. Recognizing these differences aids in choosing appropriate regression techniques, ensuring accurate interpretation of results, and effectively addressing the research questions.

Cross-Sectional Data: This type of data captures information at a single point in time across various units, such as individuals, companies, or regions. It can be understood as a snapshot, providing a diverse view across a sample at a given moment. The strength of cross-sectional data lies in its ability to compare different units simultaneously, but it doesn't track changes over time. (Miller and Yang 1998)

Time-Series Data: In contrast, time-series data tracks the same unit across different time points. It's like a series of snapshots taken at regular intervals, revealing trends and patterns over time. This type of data is valuable for analyzing temporal effects and forecasting, but its focus is limited to one unit or a homogeneous group. (Miller and Yang 1998)

Panel Data: Panel data merges the dimension of time-series with the dimension of cross-sectional data, observing multiple units over time (Markus 1979a). It's classified as either balanced, where each unit is observed at all time points, or unbalanced, with units having varying numbers of observations. Unbalanced panels present complexities in handling and interpretation. Generally, panel data allows for more nuanced analyses, capturing both individual and temporal variations (Baltagi 2021).

The dataset that is analyzed in the scope of this paper, containing programming activity and economic indicators across NUTS-3 regions from 2008 to 2020, classifies as panel data due to its temporal and spatial dimensions. It combines time-series data, tracking changes over the years, with cross-sectional data, comparing different regions simultaneously. The dataset is balanced because there is an equal number of observations for each region across the same period, from 2008 to 2020.

7.2.2 Regression Methods for Panel Data

There are different regression methods that can be used in order to estimate panel data. One possible approach is to estimate a regression model for each region in the panel data, using time-series data for that region. This captures the temporal dynamics of each region but fails to utilize the information across different regions. The major drawback is the inability to analyze or compare the variations between different cross-sectional units, as it focuses solely on time variations within each region. This approach results in disparate pieces of information, making it challenging to comprehensively assess how the independent variables influence the dependent variable. In contrast, cross-section regression is conducted at a single time point, treating each region as an observation. This method is useful for understanding differences between regions at a specific time but disregards the temporal dimension. Its limitation lies in not capturing the time-related changes or trends within each region, thus missing out on the dynamic aspects of the data.

Each of these methods, when applied to panel data, has inherent limitations. They either ignore the cross-sectional differences (in time-series regression) or overlook the temporal dynamics (in cross-section regression), failing to fully capture the structure of panel data. This leads to a potential loss of valuable information and insights that could be derived from a more comprehensive analysis considering both dimensions.

Panel data regression combines the strengths of time-series and cross-sectional analyses, capturing the dynamics of individual behavior over time while also accounting for differences across regions. The general formula for panel data regression can be represented as follows (Bell and Jones 2014):

$$Y_{it} = \beta_0 + \beta_1 X_{it} + \varepsilon_{it}$$

where Y_{it} is the dependent variable, X_{it} the independent variable, β_0 the intercept, β_1 the coefficient, and ε_{it} the error term for each region i in year t.

The panel data regression models that are used for the purposes of this research are defined in the following chapter.

7.3 Methodology

The methodology section outlines the approach for analyzing the panel dataset. This includes applying linear regression models suitable for panel data (Pooled OLS) in order to predict economic growth. The first step of the analysis compares the ability to explain programming activity based on the economic factors GDP per capita, employment and population between the initial linear model and different Machine Learning models (Random Forest, Gradient Boosting, Neural Networks). The main criteria for comparison are R-squared, which measures how much variance in programming activity is explained by the given economic indicators, and Mean Squared Error (MSE), used specifically for evaluating the ML models. Similar to the approach in the main part of the dissertation, the excess programming literacy, which is described by the residuals in predicted programming activity, is used to forecast growth in GDP. However, instead of performing this process on just one cross-section (2016), this time the whole panel data is used, essentially expanding the scope of the analysis.

This approach is guided by several research questions: What additional insights can we gain from using the whole panel data in our models? Are ML models able to capture complex, non-linear relationships more effectively in the data, and if so, can they better explain programming activity based on economic factors than linear regression models? Lastly, the research will explore whether programming activity can essentially be understood as a valuable metric that could help to eventually make more reliable predictions of future economic growth.

7.4 Explaining Programming Activity based on Economic Factors

The preliminary step in this exploration is to establish a baseline using a linear Ordinary Least Squares (OLS) regression model on the whole dataset. The OLS model's summary, as shown in Table 12, indicates a moderately strong R-squared value of 0.596, suggesting that approximately 60% of the variability in programming activity can be explained by the economic indicators included in the model. When investigating the whole panel data, it can be observed that this result is significantly lower than the R-squared value of 0.868 for the baseline model of the cross section (2016) that was explored earlier in the thesis.

Table 12 Model Summary for the Baseline OLS Regression

Statistics:	**Value:**		
R-squared:	0.596		
F-statistic:	1944		
Prob (F-statistic):	0.00		
	coef	**std err**	**P>|z|**
const	-1837.2685	205.043	0.000
EMP (THS)	37.7955	0.990	0.000
GDP_per_cap	0.0656	0.006	0.000
POP (THS)	-12.2390	0.482	0.000

GDP per capita shows a positive correlation with programming activity, as does employment, while population presents a negative relationship. The F-statistic as well as the independent variables' coefficients are statistically significant with a p-value of 0. These results set a precedent for the subsequent analysis using machine learning (ML) algorithms.

The next phase involves the application and evaluation of various ML algorithms to determine if they can offer enhanced predictive power compared to the baseline linear model. ML algorithms, like Random Forests or Neural Networks, can potentially offer better explanations and results, especially when the relationships between variables are complex and not adequately explained by linear models. They are very good at capturing complicated patterns and dependencies (Najafabadi et al. 2015). Additionally, some ML models can automatically detect and model interactions and non-linearities without explicitly specifying them (2015). This flexibility and robustness can make them valuable for uncovering deeper insights and achieving higher predictive accuracy. However, regarding interpretability with machine learning models, the complexity and often "black-box" nature of these models can make it difficult to understand how they arrive at predictions or to interpret the specific influence of each variable (Carvalho, Pereira, and Cardoso 2019). This contrasts with the linear regression model addressed before, which generally offers clearer insights into the relationships between variables.

The metrics Mean Squared Error and R-squared are used in order to compare the performance of the different models. To prevent overfitting, a hold-out validation set is created to test the best models found on unseen data. For this, a proportion of data (20%) is randomly sampled from each region to create the hold-out set. This method ensures that the validation set is representative of the entire dataset across all regions. Additionally, two primary steps of feature engineering are implemented: creating region-specific dummy variables and adding a time trend variable. The region-specific dummies are introduced to capture the unique characteristics

and influences of each NUTS-3 region, transforming categorical regional data into a numerical format that the model can process. Meanwhile, the time trend variable is included to represent the chronological progression. This treatment of the temporal dimension is crucial in panel data analysis as it helps to capture trends and other time-related dynamics that might influence the dependent variable.

The goal of this analysis is to select the best-performing model, which will not only provide insights into the dynamics of programming activity in relation to economic factors but will also be pivotal in forecasting future economic growth by analyzing the residuals—the differences between observed and predicted values of programming activity.

7.4.1 Random Forest Regression

Random Forest Regressors are a type of ensemble learning method used for regression tasks, where the output is a continuous value. During training, they build many different decision trees and output the average prediction of each tree individually. This approach helps in dealing with overfitting, common in decision tree models (Breiman 2001). The Random Forest model is known for its robustness (Roy and Larocque 2012) and ability to handle non-linear relationships (Auret and Aldrich 2012).

A critical aspect of leveraging Random Forest effectively is the fine-tuning of its hyperparameters, which significantly influence the model's performance. To this end, a grid search approach is employed with cross-validation, utilizing the *GridSearchCV* module from scikit-learn. To prevent overfitting, cross-validation involves splitting the dataset into numerous subsets, training the model on some of them, and testing it on others (Jabbar and Khan 2015). This process provides a more generalized performance metric, as it assesses the model's predictive capability across various data segments. The Grid Search method systematically explores a range of hyperparameter values, identifying the optimal combination for the model.

The parameters tuned include the number of estimators, tree depth, and parameters that control tree growth.

Table 13 Summary of the Random Forest Regression Model results after Grid Search

Statistics:	**Value:**
MSE Score	8,352,094.25
R-squared	0.874
Best Model Parameters	max_depth: 10, min_samples_leaf: 4, min_samples_split: 2, n_estimators: 300

With an R-squared value of 0.8745 (see Table 13) the Random Forest model performs notably better than the baseline linear regression model discussed before, which implies that the model accounts for a larger percentage of the variation in programming activity.

7.4.2 Gradient Boosting Machine

In the pursuit of further improving the predictive accuracy of the panel data analysis, focus shifts from a Random Forest Regressor to a Gradient Boosting Machine (GBM) model. GBMs are strong machine learning algorithms that can be applied to tasks involving both classification and regression. Base learners, usually decision trees, are built sequentially and each one tries to reduce the bias of the combined learner. The motivation is to integrate several weak models to produce a powerful ensemble (Han 2023a, 33). GBMs adjust the weights of instances based on the gradient of the loss function, hence the name. This technique can result in improved predictive accuracy (Natekin and Knoll 2013). However, Gradient Boosting will usually continue improving the model to minimize all errors. This can overemphasize outliers and cause overfitting, which is why cross-validation has to be implemented to neutralize this effect (Han 2023a, 62). Again, feature-engineered variables that encode regional and temporal characteristics are included in the model.

Table 14 Summary of the GBM Model results after Grid Search

Statistics:	**Value:**
MSE Score	4,859,594.48
R-squared	0.927
Best Model Parameters	learning_rate: 0.1, max_depth: 3, min_samples_leaf: 2, min_samples_split: 2, n_estimators: 300

The results (see Table 14) are remarkable: the GBM model achieves an MSE of 4,859,594.48 and an R-squared value of 0.927. These metrics indicate a substantial improvement in predictive accuracy and model fit compared to the initial Random Forest model. The GBM's superior performance could be attributed to its ability to build trees sequentially, where each tree learns and compensates for the errors of its predecessors, a process not employed by Random Forest. This sequential, error-correcting approach allows GBM to be more flexible and adaptive, especially when dealing with complex interactions and non-linear relationships in the data (Natekin and Knoll 2013).

In conclusion, the application of the GBM model, with carefully tuned hyperparameters, demonstrates a substantial enhancement in explaining programming activity in NUTS-3 regions based on the economic indicators. The model's performance, as seen by the lower MSE and higher R-squared values, surpasses that of the Random Forest model, indicating that GBM could be a more powerful and effective tool in this context.

7.4.3 Neural Networks

Inspired by the human brain, neural networks are composed of a layer of artificial neurons (also known as "nodes"), where each connection represents a weighted influence on the signal passing through it (Han 2023b, 6). They are designed to recognize patterns and make decisions based on input data. Usually each input is separately weighted, and the sum is passed through a nonlinear function known as an activation function or transfer function (2023b, 8). Neural

networks learn from data through a process called training, adjusting these weights to optimize performance (Abdi, Valentin, and Edelman 1999).

The neural network, designed using *TensorFlow* and *Keras*, follows a sequential architecture. The model begins with a dense input layer whose number of neurons is tuned between 32 and 256. The architecture includes between 1 to 4 hidden layers, each with 16 to 128 neurons, determined by the Keras tuner *RandomSearch*. This hyperparameter optimization process evaluates 10 different model configurations to minimize validation loss, with each configuration tested once. The hidden layers use a ReLU function to deal with non-linear relationships, while the output uses a linear activation function suitable for regression tasks. The model's optimizer is Adam, with a learning rate that varies logarithmically between 0.0001 and 0.01, and it uses MSE to measure prediction errors during learning. Prior to training, the feature set is scaled using a *StandardScaler* to normalize the data. This ensures that each feature is weighed properly by the model. Once the best model is identified, it is trained for 50 epochs with a batch size of 32.

Table 15 Summary of the Neural Networks Model results after Random Search

Statistics:	**Value:**
MSE Score	1,388,690.53
R-squared	0.979

Upon evaluation, the neural network model exhibits an MSE of 1,388,690.53 and an R-squared value of 0.979 on the hold-out test set (as seen in Table 15). These metrics indicate a high level of predictive accuracy and a strong fit to the data, surpassing the performances of both the Random Forest and Gradient Boosting models. The superior performance of the neural network model could be attributed to its ability to capture complex and non-linear relationships in the data more effectively than the other models (Xinli et al. 2018). Due to their deep architectures

and non-linear activation functions, neural networks might therefore be a good choice when analyzing this kind of data.

7.4.4 Model Selection

The analysis shows that the neural networks model performs best in explaining programming activity on the whole panel data. With an MSE of 1,388,690 on the test set and an R-squared value of 0.979 it outperforms the other models, including the linear baseline model. However, neural networks are prone to overfitting the underlying data. An R-squared value of 98% seems unreasonably high. Additionally, the goal of this part of the analysis is to use the best identified model to determine excess programming literacy by calculating the difference between the actual programming activity and the predicted values. These residuals are expected to give us insights when predicting future economic growth. Therefore, diminishing them more and more questions their usability in further steps of the process. Future research could investigate whether the fact that ML models seem to explain programming activity very well could make using residuals in the further regression obsolete. In conclusion, the gradient boosting machine is chosen as the best model and the residuals are calculated accordingly.

7.5 Predicting Future Economic Growth

The focus of this part of the analysis is on determining the influence of programming activity on economic growth. In this case, the target variable of the regression models will be defined as the 3-year lagged absolute growth in GDP per capita. The effect of excess programming literacy, representing the variance in programming activity not explained by the economic indicators, is a central element in the prediction strategy. These residuals are derived from the GBM model identified as the best performing predictor in the last chapter. This approach involves creating two sets of forecasts for economic growth: one that incorporates these residuals and another that does not, allowing for a comparative analysis of their predictive

power. The analysis is set to explore the Pooled Ordinary Least Squares model to ascertain its efficacy in capturing the dynamics of economic growth. Through this methodology, the chapter seeks to assess and contrast the models' ability to harness both the visible and latent factors influencing economic development.

Before starting with the regression, the necessary variables need to be defined. The goal is to predict economic growth based on programming activity. Therefore, the dependent variable is set as the 3-year lagged growth in GDP per capita (*GDP_3y_growth)* and the independent variable as the programming activity (*activity*). Additionally, the values of GDP per capita and population of a given year are included as independent variables, because one could argue that a region that already experiences economic flourishing tends to grow accordingly. For the second run of the model, the excess programming activity (*activity_eps*) is going to be included to investigate its effect on the regression. When running the models, the following results can be observed:

Table 16 Results of the Pooled OLS Regression Model with and without Residual

Statistics:	**Value:**	
	no Residual	with Residual
R-squared:	0.3719	0.3720
F-statistic:	599.23	449.48
Prob (F-statistic):	0.0000	0.0000

Table 17 Independent Variables Statistics based on the Pooled OLS Regression with and without Residual

	coef		**std err**		**P>|z|**	
	no Residual	with Residual	no Residual	with Residual	no Residual	with Residual
const	-2143.2	-2137.0	114.22	114.55	0.0000	0.0000
activity	-0.0251	-0.0241	0.0079	0.0080	0.0014	0.0026
activity_eps		-0.0838		0.1155		0.4681

GDP_per_cap	0.1184	0.1182	0.0031	0.0031	0.0000	0.0000
POP (THS)	-0.1480	-0.1546	0.0852	0.0857	0.0823	0.0712

Table 16 and 17 summarize the results when performing Pooled OLS regression on the panel dataset. The R-squared values, indicative of the models' explanatory power, are virtually identical, with a marginal increase from 0.3719 to 0.3720 when including residuals. This suggests that the residuals add minimal explanatory benefit to the model. The F-statistics, reflecting the models' overall significance, remain highly significant ($p < 0.0001$) in both scenarios, although the value decreases from 599.23 to 449.48 with the inclusion of residuals, implying a slightly better fit without them.

Looking at the coefficients, *activity* has a negative coefficient in both models (-0.0251 without residuals and -0.0241 with residuals), and it is statistically significant ($p = 0.0014$ without residuals and $p = 0.0026$ with residuals). This suggests that an increase in programming activity is associated with a small but statistically significant decrease in GDP per capita growth. The *activity_eps* coefficient (-0.0838) when included is not statistically significant ($p = 0.4681$), indicating that the residuals from the previous model do not provide a significant contribution to predicting GDP growth in this model. The *GDP_per_cap* variable shows a positive and highly significant coefficient (0.1184 without residuals and 0.1182 with residuals), meaning that for each euro increase in GDP per capita, the model predicts an additional increase of approximately 0.12 euros in the 3-year lagged GDP per capita growth, holding all else constant. *POP (THS)* has a negative coefficient, which becomes slightly more negative when residuals are included (-0.1480 without residuals and -0.1546 with residuals), and it is also statistically significant ($p = 0.0823$ without residuals and $p = 0.0712$ with residuals), indicating that an increase in population is associated with a decrease in GDP per capita growth.

In summary, the addition of residuals to the model does not substantially improve the predictive accuracy for future economic growth. Programming activity has a small but significant negative effect on GDP per capita growth. GDP per capita and population are significant predictors with GDP per capita having a positive effect and population a negative effect on the economic growth indicator. However, when using Pooled OLS regression for panel data, there are several assumptions and limitations to consider. The regression model assumes that the relationships between the independent and dependent variables are consistent across all regions and over time (Dielman 1983). This means that it does not account for possible variations in individual regions or over different time periods. Furthermore, if there are unobserved individual characteristics that both affect the dependent variable and are correlated with the independent variables, Pooled OLS may result in biased and inconsistent estimates (1983). This method also assumes no autocorrelation (the residuals are not correlated across time) and constant variance of errors (homoscedasticity) across all panels (1983). Violations of these assumptions can lead to inefficient and unreliable estimators (McManus 2015). All these considerations indicate that this kind of regression method is not always suitable for panel data.

7.6 Conclusion

The exploration of ML models provides significant insights into the relationship between economic factors and programming literacy. They outperformed the linear regression model in explaining programming activity. The standout performer was the neural networks model, achieving an impressive R-squared value of 0.98, a substantial improvement over the linear models' average of 0.979, and an MSE of 1,388,690.53. However, it is crucial to acknowledge the potential of overfitting in ML models, a common challenge in such advanced computational techniques, which is why the GBM model was chosen for the following steps of the analysis. Although precautions like using a hold-out validation set and implementing cross-validation

were taken, the reliability of these exceptionally high-performance metrics remains a question for further investigation.

Future research could investigate whether the implementation of more and more advanced ML algorithms could make the use of residuals of explained programming activity less important. Regarding the prediction of future economic growth, a deeper statistical analysis using different linear regression models suitable for panel data, like Fixed Effects and Random Effects models, could enhance the findings and interpretations made in this paper. This approach could provide a more nuanced understanding of the dynamics at play and offer a comparative perspective to the Pooled OLS models' results.

8 Analysis of Programming Literacy and Economic Growth in Poland (2008-2020)

Previously we have examined programming literacy in countries of Europe: France, Sweden, Switzerland, Spain, and Italy that have shared a similar economical stage of growth (Eurostat 2023). Due to its distinctive economic landscape and technological development, Poland was included in the analysis for diversification. As a result, we are trying to answer what are the main factors that define the relationship between Programming Literacy and Economic Growth in Poland and how those are different from the previously analyzed countries. . Poland's economic status and technological advancements offer a distinct perspective, enriching our understanding of the interplay between programming skills and economic prosperity. Through the exploration of this additional layer, we gain valuable insight into potential variations and contribute to our research.

8.1 Mapping Poland poster location to NUTS-3 Regions

Crucial preparatory step involves the mapping of Polish towns and cities to larger NUTS-3 Regions. NUTS-3 regions, being a fundamental statistical unit in the European Union for data

analysis, serves as the basis for evaluating economic, financial, and social indicators. This process was undertaken to enhance data privacy for Stack Overflow users and facilitate the integration of Stack Overflow data with Eurostat's economic indicators reported at the NUTS-2/3 level.

The methodology involved several key steps. In the initial phase of this study, the collection of data involved obtaining information from Stack Overflow on poster locations in Poland, complemented by a dataset detailing NUTS within the country. Subsequently, as part of the preparation for mapping, each town or village in the Stack Overflow dataset underwent the assignment of the corresponding NUTS-3, NUTS-2, or NUTS-1 code. The intricacy of towns sharing names across multiple regions mandated a nuanced approach, considering both town names and their associated regions. The subsequent phase focused on data matching, where a mapping Table was crafted using the dataset containing information about NUTS regions in Poland, exported from the Central Statistical Office of Poland. This Table served as a key component in associating towns with their respective NUTS regions. The intricacies of this process lay in addressing discrepancies between town names written in polish language and English, as well as tackling instances of misspelled or incomplete town names.

An investigation of remaining unmatched locations was conducted as part of the finalization and validation phase. This investigation delved into potential issues such as misspelled or incorrectly typed locations. A new column was introduced, housing correct names while retaining the original names as identifiers. This comprehensive methodology not only preserved data integrity but also laid the groundwork for robust analysis.

8.2 Analysis of programmers' density in Poland per Population and GDP

After extracting Stack Overflow data specific to Polish regions, the next phase involved integrating it with additional socio-economic factors such as Gross Domestic Product, Salary,

Employment, Population, Number of Students, and more. Following a similar approach to previous countries, the crucial initial step was defining programmers based on Stack Overflow Survey Developer data and aligning them with discussion participation insights derived from NUTS-3 regions. The study further investigates correlations among programmer density, economic indicators, and regional characteristics, providing an intricate perspective on Poland's technological landscape. The research concludes with a predictive model, highlighting the correlation between programming density and key economic factors to offer a comprehensive understanding of regional programming activity.

8.2.1 Clustering programmers

The main dataset from Stack Overflow, contains questions and answers submitted by users on the main forum. Recognizing that each programmer interacts with Stack Overflow multiple times in their professional lifespan necessitates a specific approach in quantifying the number of programmers. Leveraging the Stack Overflow Developers Survey 2021 as a reference, analysis based on two specific data points: participation activity and professional branch of the user.

To understand those relations, K-modes clustering was introduced, a statistical method suited for categorical data analysis (Chaturvedi et al. 2001). This technique allows us to define specific personas based on diverse attributes, offering a more personalized understanding of user behaviors within each professional branch. The modeling with K-modes clustering to our data identified three clusters: a diverse group with varying participation, individuals from different branches engaging less frequently, and a smaller, active group mainly comprising students and hobbyists. We categorized them into three programmer types: Experts, characterized by answering and occasional participation; Observers, rarely engaging in discussions; and Learners, including beginners actively participating by asking questions. Translating these

personas into our dataset, we aim to estimate programmer numbers in NUTS-3 regions based on Stack Overflow data. We established criteria for each persona, considering answer count for Experts (answers questions), excluding Observers, and for Learners (ask questions). Interestingly the personas defined are different from the results extracted on the sample for other European countries, which reflects different habits of users across countries.

8.2.2 Short Analysis of top locations

Clustering results from the previous step has helped us derive initial results suggesting a strong correlation between the cities with the highest number of programmers, employment numbers, and GDP. The major cities, including Warsaw, Wroclaw, Poznan, and Trojmiejski (their localization for reference on Figure 19), consistently top in these categories. This alignment underscores the central role of these urban centers in driving both economic output and employment, particularly in the tech sector.

Figure 19 Map of Cities in Poland.

Note. Adopted from The World Factbook (n.d.)

The further analysis reveals a consistent distribution of GDP alongside Programming Density in Poland, with the majority of regions maintaining similar GDP levels. Notably, a subtle trend emerges, indicating a potential correlation between growing GDP and increasing programming density. This suggests that regions with higher economic output also tend to exhibit higher programming activity. Analyzing the top 10 regions in programming density (ratio of programmers against the total population) over nearly a decade, we observe stability in programmer density, implying either consistent programming activity or a growth in population and activity levels. Post-2015, a slight overall increase in programming density is recorded, with Warsaw, Trójmiasto, Wrocław, Poznań, and Tarnow.

Figure 20 Top 10 Regions based on the Total Number of Programmers across the Years

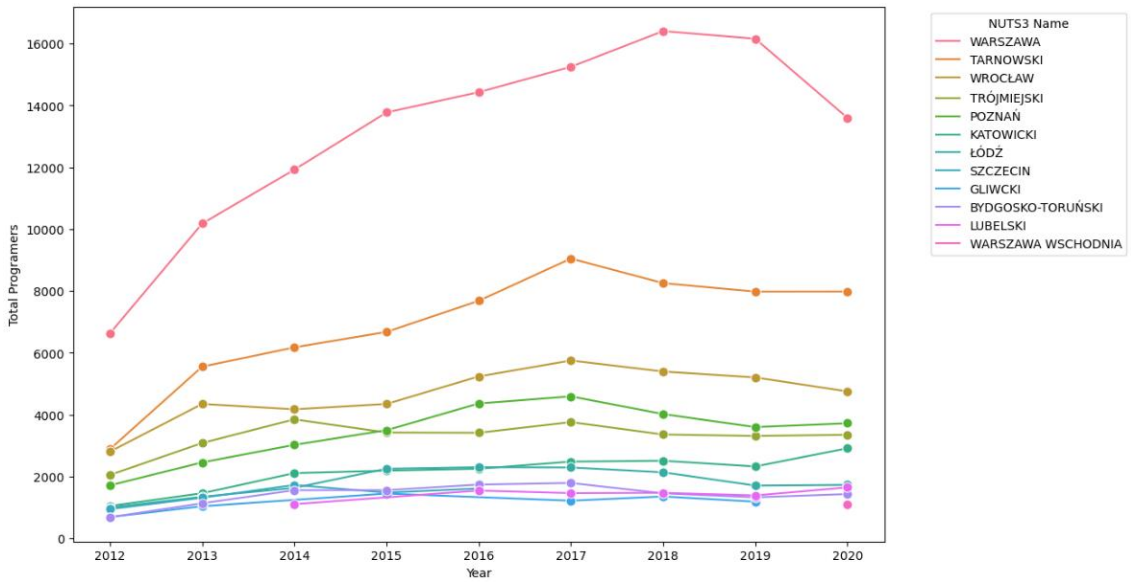


Figure 20 presents further investigation into total programmers per region underscores Warsaw's leadership, expected given its status as Poland's largest city. Surprisingly, smaller cities like Wrocław and Poznań have higher density of programmers per GDP than Warsaw, suggesting these cities not only exhibit high programming activity but also contribute significantly to regional GDP. This nuanced relationship between programming activity, GDP, and population sheds light on the dynamic technological growth experienced by Poland. As depicted at Figure 21 and Figure 22, a trend emerges in the relationship between the number of

students and the total count of programmers. It is apparent that as the number of students increases, there is a corresponding growth in the total number of programmers. Mean salary does not present as strong connection, as visible on Figure 21. However, this correlation is not particularly strong. Upon narrowing our focus to the top locations in terms of programming activity, it becomes evident that these regions often exhibit a robust presence of universities.

Figure 21 Distribution of Mean Salary compared with Total Programmers

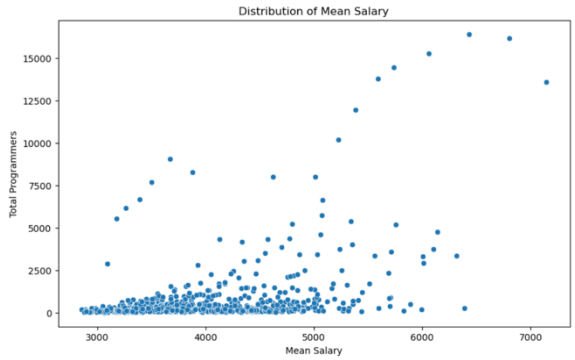
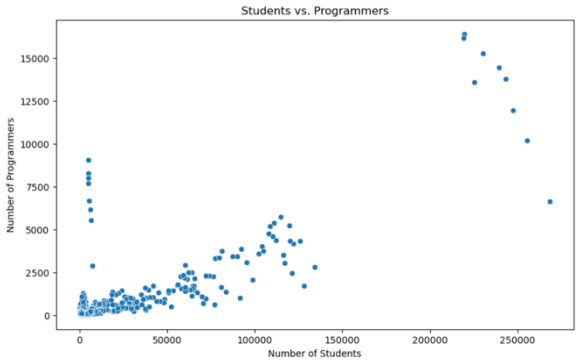


Figure 22 Distribution of Number of Students compared with Total Programmers



Unexpectedly, when comparing this with the correlation between the total number of programmers and the count of tech universities, the relationship weakens. This suggests that students enrolling in universities to learn coding may not have a strong connection to the subsequent programming activity. This discrepancy could be attributed to factors such as students not actively participating in Stack Overflow discussions, which serve as our primary indicator of programming activity, or tech universities being situated in regions lacking a developed professional landscape in technology.

Conversely, when examining the mean salary across all regions, there appears to be a lack of strong correlation with programming activity within those regions. The distribution of mean salary seems more random, reflecting a closer association with overall employment trends. This observation implies that in regions with higher job opportunities, where more companies are present, salaries tend to be higher. Additional conclusions suggest the need for a nuanced understanding of regional dynamics, including the influence of university presence and

economic factors, to comprehensively interpret the relationship between educational pursuits, employment opportunities, and programming activity

8.3 Predicting programming activity in Poland with the use of economic and social factors

8.3.1 Initial OLS Regression Model

This chapter tries to explain the relation between economic factors and programming literacy to understand whether those factors can predict programming literacy in future. In comparison to the analysis of the previous countries, for Poland it was possible to provide more indicators on NUTS-3 level: mean salary of employed people, amount of students, ratio of the population living in the cities, number of technical-focus universities. Additional variables will be used for an enriching prediction model to build a more stable and generalized model. To grasp the relationship between variables and their impact on predicting programming activity in Poland, a correlation analysis was conducted. Key features were selected as follows: Gross Domestic Product, Mean Salary, Students, and Employed as predictors for programming literacy.

Ordinary Least Squares (OLS) was chosen as the initial model—a statistical method in regression analysis aiming to estimate the relationship between a dependent variable and independent variables. The general formula for OLS Regression can be represented as follows:

$$Y = \beta_0 + \beta_1 X_1 + \ldots + \beta_n X_n + \varepsilon$$

- Y represents the dependent variable (the variable you are trying to predict).
- β_0 is the intercept term, representing the value of when all the independent variables (X_1,X_2, …, X_n) are zero.
- (β_0 , β_1 , …, β_n) are the coefficients of the independent variables, representing the change in Y for a one-unit change in the corresponding X.
- (X_1,X_2, …, X_n) are the independent variables.
- ε is the error term, representing the unobserved factors that affect Y but are not included in the model.

The goal of OLS regression is to estimate the values of (β_0 , β_1 , …, β_n) that minimize the sum of squared differences between the observed values of Y and the values predicted by the model. This is achieved by minimizing the sum of squared residuals. This method is suitable for predicting continuous outcomes based on linear relationships, assuming certain underlying conditions like linearity, independence of errors, and homoscedasticity (Mosteller and Tukey 1977). White's heteroskedasticity-robust standard errors were used in the model in order to prevent heteroskedasticity. Heteroskedasticity refers to the situation where the variability of errors is not constant across all levels of the independent variable, and it can lead to inefficient and biased standard errors. (White 1980). The Ordinary Least Squares (OLS) regression reveals a relationship between the dependent variable - programming activity and independent variables, as presented in Table 12.

*Table 12 I*ndependent Variables Statistics based on the OLS Regression

	coef	**std err**	**P>|z|**
const	5953.0309	2425.154	0.000
EMP (THS)	-19.2352	4.099	0.000
MIO_EUR	0.7180	0.092	0.000
STUDENTS	0.0919	0.010	0.000
MEAN_SALARY	-1.3016	0.509	0.011

Upon conducting an initial examination of the variables p-value in Table 12 of the OLS Regression outcomes at the individual variable level, it becomes evident that a considerable number of variables exhibit statistical significance specifically for Poland. In comparison to a parallel model executed at the European level, which encompasses multiple countries, the significance levels of the variables are notably elevated. This implies that constructing models at the country level, as opposed to a broader regional level, may yield more advantageous results

for accurately predicting programming literacy. Notably, the inclusion of additional variables such as mean_salary (average salary) and students (number of students) has a discernible impact on the predictive accuracy. This assertion is supported by the overall model statistics outlined in Table 13.

Table 13 Results of the OLS (Ordinary Least Squares) Regression Model

Statistics:	**Value:**
R-squared:	0.810
Adj. R-squared:	0.805
F-statistic:	158.3
Prob (F-statistic):	2.46e-50

The R-squared value measures the percentage in which dependent variables can be explained by others. Compared to the model performed in five countries, ours is substantial at 81.0%. Higher R-squared value suggests a better fit, showcasing the model's effectiveness in explaining the variability of our target value, in this case programming literacy. However, it's essential to note that while the model demonstrates a strong explanatory power, careful consideration of individual coefficient interpretations, and large condition number stating the level of features multicollinearity, is warranted for a comprehensive understanding of the model's performance. The analysis of the residuals offers valuable insights into the performance of the regression model. The mean being close to zero suggests that, on average, the model captures the underlying patterns in the data, minimizing systematic bias. However, the relatively large standard deviation of 2.29e+03 indicates significant variability in individual prediction errors, implying that while the model performs well on average, there are instances where predictions deviate substantially from actual values. Considering that it is necessary to explore other models.

8.3.2 Model Selection and Estimation:

In this comprehensive model comparison, various regression techniques were systematically evaluated to select the most effective model based on the R-squared (R2) score. The exploration extended to regularized linear models, including Ridge and Lasso Regression, to increase generalization and several Machine Learning Models. The subsequent analysis delved into non-linear models, introducing Random Forest, Decision Tree, Gradient Boosting, Support Vector Machines (SVM), Neural Networks (MLPRegressor), and XGBoost into the comparison. The results comparing R2 of all models tested are presented in Table 14.

Table 14 Comparison of performance of multiple predictive models

Predictive Model	**Cross Validation Score**
Linear Regression	0.810
Lasso Regression	0.897
Ridge Regression	0.896
Random Forest Model	0.334
Decision Tree	-0.227
Gradient Boosting Regressor	0.205
Neural Networks (MLPRegressor)	-0.118

Analysis of Table 14 led to the conclusion that linear regression models are on average performing better than non-linear in terms of R2 Score. Including Ridge and Lasso Regression outperformed simple regression , due to its ability to generalize relationships between variables of models. Machine Learning, non-linear models, has severely underperformed, presenting zero ability to explain the variability, and inability to predict programming literacy based on economic factors.

The findings highlighted the diverse characteristics of each model, emphasizing the trade-offs between complexity, interpretability, and predictive accuracy. Based on the model comparison

Lasso model is a suitable choice for predicting economic growth due to its ability to perform automatic variable selection, handle multicollinearity, and prevent overfitting. By introducing a regularization term that encourages sparsity in the model, Lasso identifies and prioritizes the most relevant predictors while excluding less impactful variables (Ranstam and Cook 2018). This not only simplifies the model but also enhances its generalizability to new data.

8.3.3 Prediction of programming activity based on the best model

In the analysis, an additional variable named "growth" based on the past 5-year period from 2012 to 2017 was introduced to capture the historical trajectory of economic development. This variable represents the change in GDP per capita from five years prior to the target year. The rationale behind this addition is to account for the temporal aspect of economic growth, allowing the model to consider the momentum or stagnation in economic conditions over a relevant time frame. The results from the first model, excluding the programming activity residuals, revealed varying performance across the years from 2017 to 2020. The R-squared values, indicative of the model's explanatory power, ranged from approximately 0.748 to 0.837 between the years. While these values suggest a moderately good fit, the significance of individual variables varied, and the residuals displayed a considerable standard deviation.

The Lasso regression model consistently demonstrates the most promising performance across all years, showing good generalization on unseen data. This leads to the conclusion that economic growth and social factors can explain and predict programming literacy in Poland. However, the relationship is intricate; while adding the combination of education-related and social indicators like the number of students and average salary, enhances predictive accuracy.

8.4 Analysis of Outliers in Poland

Based on residuals derived from the prediction model, few outliers were discovered. After an analysis we can notice a few regions outstanding from the rest.

For example, the region of Krakowski (around the city of Cracow) shows a substantial positive discrepancy. This is likely due to the region's high employment rate and overall GDP, caused by geographical proximity to the major Polish city of Cracow further amplifying its economic influence. The model seems to see GDP as a key driver for increased programming literacy and overestimates the final value. Moreover, additional outliers, exemplified by Szczecin and Wroclaw, present an unexpectedly increased observed literacy. While compared to other regions, those cities have relatively high numbers of students. This surge can be influenced by the presence of students, signaling a more complex relationship between educational institutions and programming literacy than initially assumed. Additionally, Warsaw emerges as an outlier with marginally higher observed activity than predicted, reflecting a thriving tech sector and a robust programming community. Despite extraordinarily high values for each variable, the model demonstrated good performance in predicting literacy, standing out among other cities. On the other hand, Tarnow presents as an outlier with remarkably higher programming literacy than predicted. Despite a weaker economic position, low GDP, mean salary, and a lower number of tech universities students, the region exhibits unexpectedly high programming activity. This requires further investigation to detect unique local conditions contributing to the deviation.

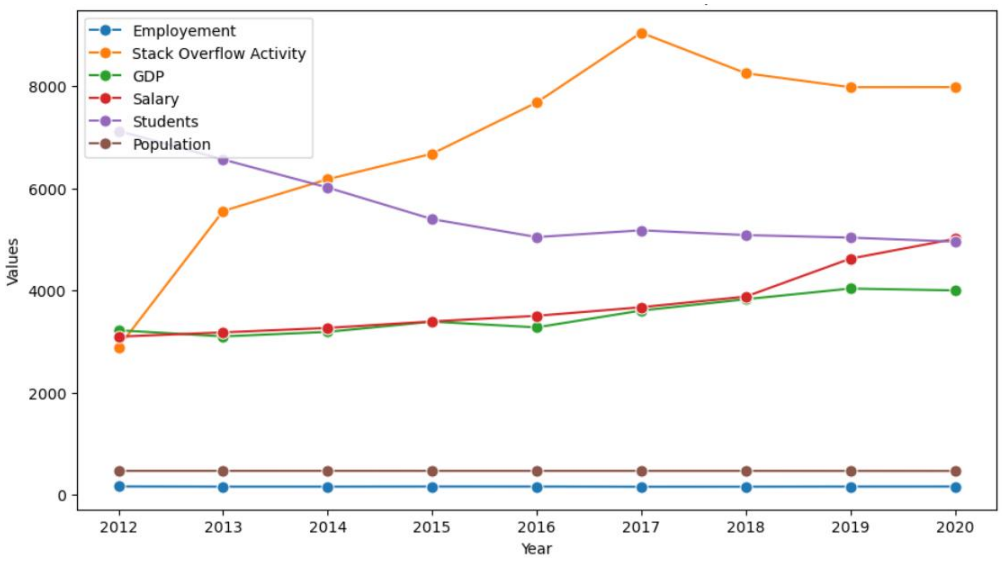
These outliers not only underscore the regional dynamics, economic conditions, and the strength of local tech ecosystems but also require exploration into the distinct characteristics of each region. This analysis has proved that programming literacy is an output of various

additional social and economic factors that would need to be included in order to understand this relation with each other.

8.4.1 Tarnow Analysis

In the case of Tarnow, the analysis results presented at the Figure 23, shows that none of the analyzed metrics offers a clear correlation with its remarkably high programming activity. Despite stagnant population and employment figures over the years, coupled with a slow and marginal growth in salary and GDP – trends more reflective of the national landscape than the region itself – the justification for the surge in programming activity remains elusive. The absence of technical universities in the area and a declining number of students aren't explaining the growth. The conventional markers of economic prosperity and educational infrastructure seem inadequate in explaining this anomaly. It beckons a more nuanced investigation into Tarnow's unique local conditions, potentially involving grassroots initiatives, industry-specific developments, or other unexplored facets that have catalyzed an unparalleled surge in programming activity.

Figure 23 Economical and Social Factors in Tarnow across the Years



To understand the cause of the growth, it is necessary to analyze other external sources. Despite a notable absence of major investments in IT, technology, or science in recent years, and a decline in the economic contribution of Grupa Azoty Tarnow, a key local company, the surge in programming activity persists (Marciniak et al. 2023). Furthermore, the region's sole university, a technical institution, records a diminishing student population over time, challenging conventional expectations of educational influence on programming activity. Yet, a compelling explanation emerges when considering Tarnow's strategic proximity to one of Poland's largest programming hubs – Cracow (refer to Figure 19). This geographical advantage reveals a plausible scenario: individuals employed in Cracow, a major tech center, choose to reside in Tarnow, contributing to the observed heightened programming activity (Małochleb et al. 2020). This unique interplay of geographical convenience and regional dynamics provides a fresh lens through which to comprehend Tarnow's outlier status, emphasizing the complex and multifaceted nature of regional programming dynamics. This nuanced understanding not only enriches our exploration of Tarnow's anomaly but also underscores the importance of considering external factors and spatial relationships in unraveling complex regional phenomena.

8.5 Conclusions

In exploring the relation between programming literacy and economic growth, this study solely focuses on Poland. Analyzing programmers' density per population and GDP identifies key programming hubs like Warsaw, Wroclaw, Poznan, and Trojmiejski, highlighting a strong correlation between high programmer numbers, employment figures, and GDP in cities. Examining GDP distribution alongside programming density suggests a potential correlation, indicating regions with higher economic output also show increased programming activity. Compared to other countries, this model wasn't able to predict well based only on economic

factors. That proves that in order to understand the complex relation between programming literacy and economic growth, more indicators need to be included. Incorporating more factors significantly contributes to defining a more generalized predictive model that is able to provide valuable insights independent of country or volatile external factors. More complex model has the potential to perform well and should be further explored with the addition of social indicators. Considering the size of a dataset and limited number of factors, more robust machine learning models are not able to keep prediction generalized.

In conclusion, the analysis underscores the complexity of the relationship between economic factors, education, and programming literacy in Poland. Nonetheless, it indicates an upward trend in programming literacy, particularly in less developed regions, which is an encouraging sign and should be further examined.

9 Conclusions

This research builds on prior work to significantly advance our understanding of how programming literacy contributes to economic growth, a relatively underexplored area. It emphasizes the societal value of programming skills and informs policy strategies to promote such literacy as a key asset for regional development. Further research displayed how using data at hand can be applied to conduct in-depth analysis on a single country. They investigated the complexity of the interplay in Poland, emphasizing the need for a tailored approach due to the country's unique development stage in comparison to other European countries. There is a discernible upward trend in programming literacy, particularly in less developed regions, signaling a positive trajectory, driven by economic growth, the availability of technical education.

Just as with growth metrics, the paper coherently showed weak, yet positive correlations between resilience and programming literacy. Nevertheless, the models' performances could

be improved to deliver more accurate and hence reliable results, and for enriched representativeness, more indicators of programming literacy included.

10 Discussion and Outlook

There are a few limitations throughout our study that need to be considered. Firstly, our dataset is restricted to five countries in Europe, all characterized by higher GDPs compared to other European nations. This raises concerns about the generalizability of our findings, particularly in regions with relatively lower GDPs. Another limitation lies in the representativeness of our chosen metric for programming literacy. The reliance on Stack Overflow data, capturing information about comments, answers, and questions, may not offer a fair representation of programming literacy. It is expected that a significant segment of individuals proficient in coding may not actively participate on such platforms. As a result, additional data sources need to be incorporated. Integrating information about projects uploaded on GitHub, a platform facilitating collaboration among programmers, would provide a more comprehensive perspective and generalize our findings. Additionally, exploring models that incorporate geospatial aspects is advised. This approach would contribute to the development of region-focused models, providing a better understanding of the factors influencing programming literacy in specific regions.

It is recommended that future research building on this study address the identified limitations. Upon resolving these, further investigation could meaningfully focus on predicting GDP by leveraging programming literacy as a significant variable, thereby deepening the understanding of the interplay between these two dimensions. Additionally, it is crucial to formulate policy implications drawn from our models' results wherefore here, too, future research is advised.

References

Abdi, Herve, Dominique Valentin, and Betty Edelman. 1999. *Neural Networks*. SAGE.

Agresti, A. (2002). Categorical data analysis (2nd ed.). Wiley.

Atomico, Slush, and Orrick. 2019. "State of European Tech 2019." 2019. https://2019.stateofeuropeantech.com/chapter/people/article/strong-talent-base/.

Auret, Lidia, and Chris Aldrich. 2012. "Interpretation of Nonlinear Relationships between Process Variables by Use of Random Forests." *Minerals Engineering* 35 (August): 27–42. doi:10.1016/j.mineng.2012.05.008.

Bălăcescu, Aniela. "Economic Growth and Digital Skills: An Overview on the EU-28 Country Clusters." Annals of the "Constantin Brâncuși", University of Târgu Jiu, Economy Series (2019), Issue 6, (2019).

Balanskat, and Engelhardt. 2014. "Computing Our Future Computer Programming and Coding - Priorities, School Curricula and Initiatives across Europe." *European Schoolnet*, October. https://doi.org/10.13140/RG.2.1.5029.9048.

Baltagi, Badi H. 2021. *Econometric Analysis of Panel Data*. *Springer Texts in Business and Economics*. doi:10.1007/978-3-030-53953-5.

Bell, Andrew, and Kelvyn Jones. 2014. "Explaining Fixed Effects: Random Effects Modeling of Time-Series Cross-Sectional and Panel Data." *Political Science Research and Methods* 3 (1): 133–53. doi:10.1017/psrm.2014.7.

Benitez, J., A. Castillo, L. Ruiz, X. Luo, P. Prades. "How Have Firms Transformed and Executed IT-Enabled Remote Work Initiatives during the COVID-19 Pandemic? Conceptualization and Empirical Evidence from Spain." Information & Management 60, no. 4 (2023).

Breiman, Leo. 2001. “Random Forests.” *Machine Learning 45* (1): 5–32. doi:10.1023/a:1010933404324.

Carvalho, Diogo Vieira, Eduardo M. Pereira, and Jaime S. Cardoso. 2019. “Machine Learning Interpretability: A Survey on Methods and Metrics.” *Electronics* 8 (8): 832. doi:10.3390/electronics8080832.

Chandra. n.d. “Generation of Computer Languages.” Scribd. https://de.scribd.com/document/99405085/Generation-of-Computer-Languages.

Chaturvedi, Anil, J. Douglas Carroll, Paul E. Green, and John A. Rotondo. "K-modes Clustering." Journal of Classification 18, no. 1 (2001).

CSDH. 2022. “What Is a Generation Computer Language and How Is It Used?” Computer Science Degree Hub. December 2, 2022. https://www.computersciencedegreehub.com/faq/what-is-a-second-generation-programming-language//.

Davis, Cameron, Ben Safran, Rachel Schaff, and Lauren Yayboke. 2023. “Building Innovation Ecosystems: Accelerating Tech Hub Growth.” McKinsey & Company. February 28, 2023. https://www.mckinsey.com/industries/public-sector/our-insights/building-innovation-ecosystems-accelerating-tech-hub-growth.

Dielman, T. E. 1983. “Pooled Cross-Sectional and Time Series Data: A Survey of Current Statistical Methodology.” The American Statistician 37 (2): 111–22. doi:10.1080/00031305.1983.10482722.

Eurostat. 2022. *Statistical Regions in the European Union and Partner Countries: NUTS and Statistical Regions 2021*. European Union. https://ec.europa.eu/eurostat/documents/3859598/15193590/KS-GQ-22-010-EN-N.pdf/82e738dc-fe63-6594-8b2c-1b131ab3f877?t=1666687530717.

GeeksforGeeks. 2023. “Generation of Programming Languages.” October 31, 2023. https://www.geeksforgeeks.org/generation-programming-languages/.

GitHub. 2023. “The Top Programming Languages.” The State of the Octoverse. 2023. https://octoverse.github.com/2022/top-programming-languages.

Investopedia. 2023. “What Is Economic Growth and How Is It Measured?” September 30, 2023. https://www.investopedia.com/terms/e/economicgrowth.asp.

Jabbar, H., & Khan, R. Z. 2015. “Methods to avoid over-fitting and under-fitting in supervised machine learning (comparative study)”. *Computer Science, Communication and Instrumentation Devices, 70(10.3850)*, 978-981.

Jain. 2018. “Generations of Programming Language.” IncludeHelp. June 26, 2018. https://www.includehelp.com/basics/generations-of-programming-language.aspx.

Laitsou, E., A. Kargas, and D. Varoutas. "The Impact of ICT on Economic Growth of Greece and EU-28 under Economic Crisis." In Internet of Things Business Models, Users, and Networks (2017).

Lewis-Beck, Michael S. 1980. Applied Regression: An Introduction. Beverly Hills, CA: Sage.

Markus, Gregory B. 1979a. *Analyzing Panel Data*. *SAGE Publications, Inc. eBooks*. doi:10.4135/9781412983389.

McManus, Patricia. 2015. “Introduction to Regression Models for Panel Data Analysis.” *Indiana University Workshop in Methods*, February. https://scholarworks.iu.edu/dspace/bitstream/2022/21957/1/2015-02-13_wim_mcmanus_panel_flyer.pdf.

Medewar. n.d. “Generations of Programming Languages: Advancements in Code.” Techgeekbuzz.Com. https://www.techgeekbuzz.com/blog/generations-of-programming-languages/.

Miller, Gerald J., and Kaifeng Yang. 1998. *Handbook of Research Methods in Public Administration, Second Edition*. *Routledge eBooks*. doi:10.4324/9780203908303. 283-284

Najafabadi, Maryam M., Flavio Villanustre, Taghi M. Khoshgoftaar, Naeem Seliya, Randall Wald, and Edin Muharemagic. 2015. “Deep Learning Applications and Challenges in Big Data Analytics.” *Journal of Big Data* 2 (1). doi:10.1186/s40537-014-0007-7.

Natekin, Alexey, and Alois Knoll. 2013. “Gradient Boosting Machines, a Tutorial.” *Frontiers in Neurorobotics* 7 (January). doi:10.3389/fnbot.2013.00021.

OECD 2023b. “Gross Domestic Product (GDP) (Indicator).” 2023. https://data.oecd.org/gdp/gross-domestic-product-gdp.htm.

OECD. 2023a. “Employment Rate (Indicator).” 2023. https://data.oecd.org/emp/employment-rate.htm#:~:text=Employment%20rate-,Employment%20rates%20are%20defined%20as%20a%20measure%20of%20the%20extent,to%20the%20working%20age%20population.

OECD. 2023c. “Value Added by Activity (Indicator).” 2023. https://data.oecd.org/natincome/value-added-by-activity.htm.

Pons, Antoni, and Onofre Rullan. "The Expansion of Urbanization in the Balearic Islands (1956–2006)." Journal of Marine and Island Cultures 3, no. 2 (2014): 78-88.

Roser, Max. 2023. “What Is Economic Growth? And Why Is It so Important?” Our World in Data. October 16, 2023. https://ourworldindata.org/what-is-economic-growth.

Roy, Marie-Hélène, and Denis Larocque. 2012. “Robustness of Random Forests for Regression.” *Journal of Nonparametric Statistics* 24 (4): 993–1006. doi:10.1080/10485252.2012.715161.

Sartore, Melissa. 2022. “What Is Stack Overflow? A Forum for All Who Code.” ZDNET. October 7, 2022. https://www.zdnet.com/education/computers-tech/what-is-stack-overflow/.

Schaefer, Christof. 2023. “The Programming Languages Taking over in 2024 and Beyond.” *Amoria Bond* (blog). May 9, 2023. https://www.amoriabond.com/en/insights/blog/the-programming-languages-taking-over-in-2024-and-beyond/.

Sein, Maung K., Devinder Thapa, Mathias Hatakka, and Øystein Sæbø. 2018. “A Holistic Perspective on the Theoretical Foundations for ICT4D Research.” *Information Technology for Development* 25 (1): 7–25. https://doi.org/10.1080/02681102.2018.1503589.

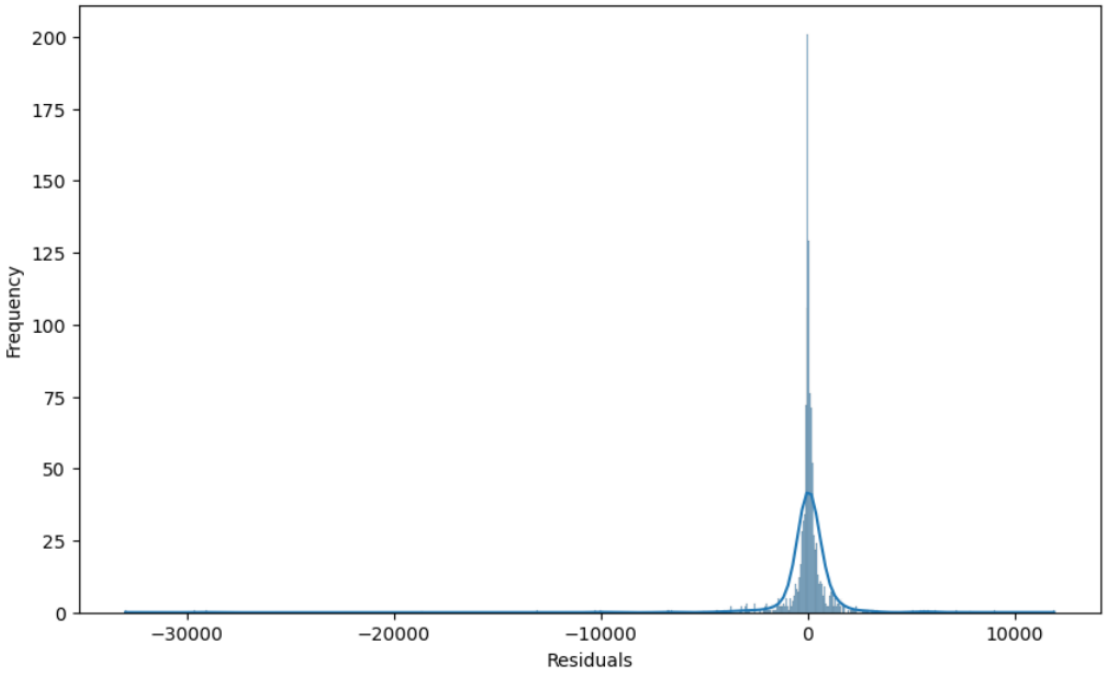
Statista. 2019. “Leading European Countries with Professional Developers in Europe in 2019, Ranked by Countries.” November 2019. https://www.statista.com/statistics/957876/professional-developers-in-europe/.

The World Factbook. n.d. “Poland - Details.” https://www.cia.gov/the-world-factbook/countries/poland/map/.

Varley, Kevin. 2023. “These Countries Are the Best at Attracting World’s Top Talent.” *Bloomberg.Com*, November 7, 2023. https://www.bloomberg.com/news/articles/2023-11-07/these-countries-are-the-best-at-attracting-world-s-top-talent.

Appendix

Appendix 1: Distribution of the Residuals based on the OLS Regression Model



Appendix 2: Links to Notebooks on GitHub:

- **Link to folder:** https://github.com/carferrom/Nova-SBE-Field-Lab
- **Analysis of Programming Literacy and Economic Growth in Poland (2008-2020):**
    - https://github.com/carferrom/Nova-SBE-Field-Lab/blob/main/Poland%20-%20mapping%20to%20NUTS2%20and%20NUTS3.ipynb
    - https://github.com/carferrom/Nova-SBE-Field-Lab/blob/main/Poland%20-%20Analysis%20of%20programmers%20density%20in%20Poland%20per%20Population%20and%20GDP%20%26%20Analysis%20of%20Outliers.ipynb